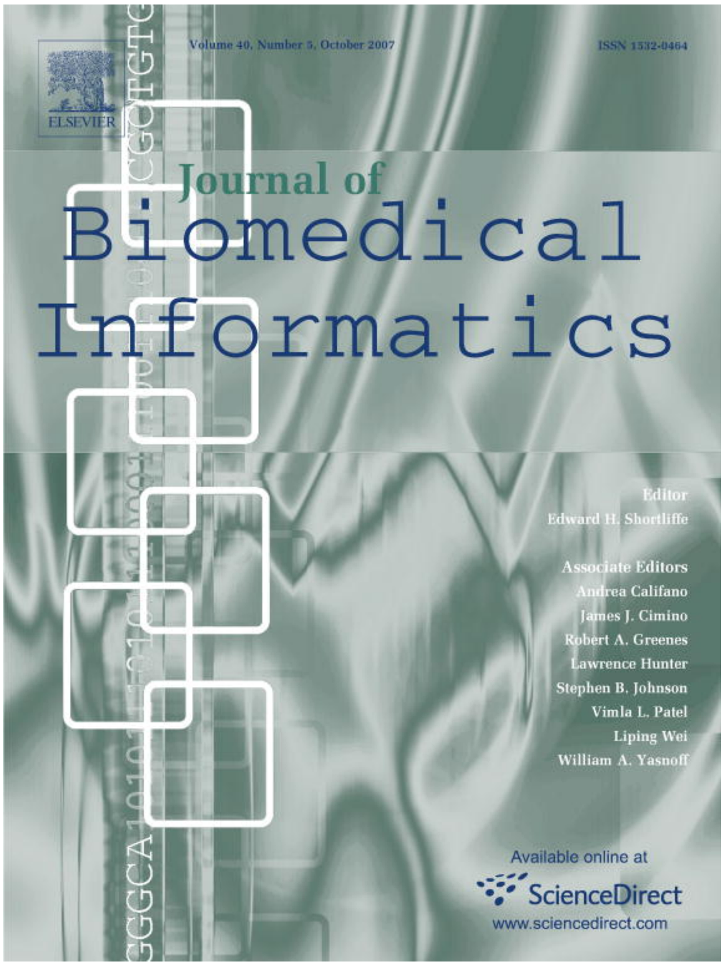
Volume 40, Number 5, October 2007
ISSN 1532-0464
ELSEVIER
Journal of
Biomedical
Informatics
Editor
Edward H. Shortliffe
Associate Editors
Andrea Califano
James J. Cimino
Robert A. Greenes
Lawrence Hunter
Stephen B. Johnson
Vimla L. Patel
Liping Wei
William A. Yasnoff
Available online at
ScienceDirect
www.sciencedirect.com

Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Journal of Biomedical Informatics 40 (2007) 486–499
Journal of Biomedical Informatics
www.elsevier.com/locate/yjbin

Interacting agents through a web-based health serviceflow management system

Giorgio Leonardi [a,*], Silvia Panzarasa [b], Silvana Quaglini [a], Mario Stefanelli [a], Wil M.P. van der Aalst [c]

[a] *Dipartimento di Informatica e Sistemistica, Università di Pavia, Italy*
[b] *Consorzio di Bioingegneria e Informatica Medica, Pavia, Italy*
[c] *Department of Information Systems, Eindhoven University of Technology, P.O. Box 513, NL-5600 MB, Eindhoven, The Netherlands*

Received 18 July 2006
Available online 19 December 2006

Abstract

The management of chronic and out-patients is a complex process which requires the cooperation of different agents belonging to several organizational units. Patients have to move to different locations to access the necessary services and to communicate their health status data. From their point of view there should be only one organization (Virtual Health-Care Organization) which provides both *virtual* and *face-to-face* encounters. In this paper we propose the Serviceflow Management System as a solution to handle these information and the communication requirements. The system consists of: (a) the model of the care process represented as a Serviceflow and developed using the Workflow Management System YAWL; (b) an organizational ontology representing the VHCO; and (c) agreements and commitments between the parties defined in a contract (represented as an XML document). On the basis of a general architecture we present an implementation in the area of Diabetes management.

Keywords: Health care; Knowledge management; Business process; Workflow; Chronic diseases

1. Introduction

1.1. Problem setting

In the past, acute diseases represented the major cause of disability for the world's population. Traditional primary care practice was largely designed to provide rapid access and care to patients with acute problems, with an emphasis on patient flow, short visits, diagnosis and treatment of actual symptoms and signs. In such a framework health-care professionals are the main actors while the patients are generally "passive subjects".

More recently, with the progressive increase of life expectancy, the management of chronic pathology and home-based care also became highly relevant problems for the health-care systems and because the associated costs now represent a relevant fraction of the total of health-care related costs.

Actual health-care systems are not adequate to meet this change [20] and the new derived issues, such as a patient management over a long period, the cooperation of personnel with different health-care skills, and the timely identification of side effects of long-term therapy. These issues result in discontinuity and fragmentation of care.

In order to cope with new needs, both practice and responsibilities are to be shared among professionals, patients and their caregivers (e.g., relatives, home assistants, etc.) [21]. In this way, far from still being a "passive subject", the patient becomes one of the main actors and the main responsible for his/her own daily management and possible behavioral changes (self-determination) [22]. This leads to a sort of "assisted self-management", a new

* Corresponding author.
E-mail address: giorgio.leonardi@unipv.it (G. Leonardi).

1532-0464/$ - see front matter
doi:10.1016/j.jbi.2006.12.002

way of care delivery that could be both efficient and effective if the organization succeeds in:

1. educational programs for spreading health-care-related knowledge among citizens;
2. retaining the patients to the institutions;
3. technology exploitation, providing patients with means for evaluating their status.

Self-management, of course, does not mean that patients take critical decisions by themselves: they will regularly need to interact with health-care organizations. Even more, the management of chronic and out-patients requires the cooperation of different actors belonging to different health-care organizations. Patients need several health services: specialist visits, instrumental examinations, therapies and prescriptions. Moreover, they may need to communicate their health status data from home or from wherever they are. The challenge is to lead them through this flow of services while hiding the complexity of the different structures.

1.2. Summary of approach

We designed and built a Serviceflow Management System, which is able to manage the overall care delivery process by establishing a tight link between different organizational units and professionals without mutual knowledge about their work processes. This goal can be archived only through an efficient and safe communication between the different actors and a synchronization of all the processes involved. The architecture proposed allows a single organization to deliver the required services and it fosters patient self-management supporting his/her journey into a complex net of services. In this paper, we describe a system that tries to meet these requirements and the corresponding implementation in the area of Diabetes management.

1.3. Outline of paper

The remainder of this paper is organized as follows. Section 2 presents the approach and the architecture model within a virtual health-care organization. Section 3 summarizes the technological solutions adopted in realizing the Serviceflow Management System (e.g., ontologies, contracts and YAWL). Next, Section 4 presents the clinical application. Conclusions are given in Section 5.

2. Approach

2.1. Virtual health-care organization

From the patient point of view there should be only one organization. All the actors involved in the care, both human and software agent, should seamlessly constitute the so called *Virtual Health Care Organization* (VHCO). A VHCO is a partnership of different health care organizations (HCOs), which can be in different places, but cooperate towards a common goal (in our case the care process management).

To reach this goal, the involved *Organizational Units* (OUs) have to share knowledge, resources, data and fragments of processes [23]. An OU represents an HCO, or the divisions of an HCO offering different services. The VHCO processes are composed of several activities distributed across several organization units. Each OU performs a part of the overall care process, supplying the patients with the necessary services. We assumed that the whole care process is managed through both *virtual* and *face-to-face* encounters.

The fundamental part of the care management is represented by home-care activities carried out by patients: self-monitoring of clinical data and self-administration of drugs. On a regular basis the overall status of the patient is examined by health care providers through periodical control visits. For planning and adapting therapy, and in addition other specialist visits, exams and tests may be scheduled (*face-to-face* encounters).

In addition to this ‘‘classical’’ care delivery, patients can also communicate any useful information about their health status via the Internet. Software agents can also ask specific questions to patients, acting as automatic monitoring system [25], such as the *Multi Access System* (MAS) described in [26]. This information is examined by the health-care providers in a synchronous or an asynchronous way. So, thanks to the remote connection via the Internet, the patient is able to send data and receive recommendations (*virtual* encounters).

In our approach, every OU offers services not covered by any other OU in the VCHO. The initial VHCO's composition must be approved by the patient, because she/he must trust the OUs in the VHCO and the way they offer the services. This composition must not change for a long time, and changes such as the integration of new OUs or the substitution of already integrated ones must be contracted and accepted by the patient and must be strongly motivated and necessary.

2.2. Architecture

The architecture of the proposed system, as shown in Fig. 1, involves three levels.

At the *Organizational Units* (OU) level, OUs manage their processes and activities using applications and/or workflow systems [30]. The overall care process involves the cooperation between the different OUs, but this doesn't have to affect the way the activities are implemented. To obtain this, each OU implements its activities with private processes, without exposing its internal structure. Other OUs just know that certain activities are carried out, without caring about how they are implemented.

To allow the communication between different OUs, fragments of processes are published through *Service Pro-*

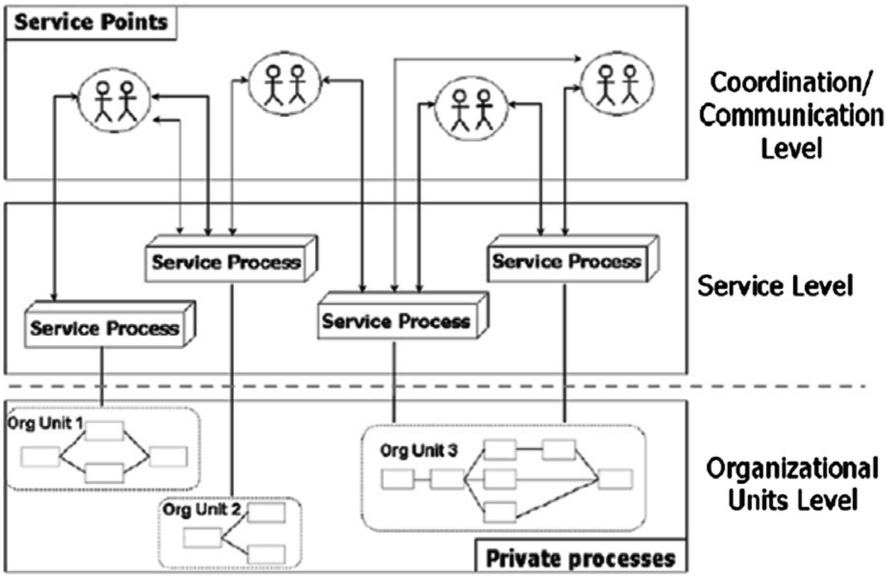

Fig. 1. System architecture.

cesses (SPRs), that act as a public interface. A SPR is a public abstraction of the services the organizational unit is able to deliver. A SPR definition contains information on the service provider and the conditions and/or constraints to be respected to request and obtain the service. The *service level* of the architecture is composed of the set of all SPRs that have been published.

At the top, we find the *coordination/communication* level of [23], composed of all the services offered by the VHCO to the patient. *Service Points* (SP) (defined in Section 2.3.1) represent these services and are able to coordinate and synchronize any interaction and data flow needed to fulfill the service, exploiting the SPRs as interfaces for the communication and cooperation between the OUs involved.

As highlighted by the dotted line in Fig. 1, the model clearly separates the service offered by the organization from its implementation, in order to meet the privacy needs and the implementation choices of every organization unit.

2.3. Definition of terms

Our modeling approach has also been inspired by recent work in the field of *computer supported cooperative work* [44,45]. Being concerned with social relations and interactions, we use *serviceflow modeling* (see Section 2.4) to bring out the different actors' perspectives and keep them involved while envisioning change and deciding about design and use of information technology within the service performance.

Before introducing the definitions of *serviceflow* and *serviceflow management system* (Section 2.4), however, it is useful to introduce some terms that allow to better understand these concepts and the whole system architecture: the notion of *Service Point* (SP) and *Service Process* (SPR).

2.3.1. Service points

To simplify matters and to enable structuring from the provider's point of view, we define a *serviceflow* in terms of *Service Points* (SPs). A service always creates some social situation, it needs a "place" [50] which frames the situation where service tasks are carried out, e.g.,

- service staff evaluating the client's concern and serving her/his needs (in these situations the client's presence may vary from being present, being present through telecommunication, or being virtually present through one of his objects or through a representation of the concerns);
- client is served by some auto-operational device (e.g., a web portal) on behalf of the service provider.

These places are called SPs.

The "service consumer" and the "service provider" meet at SP: here, based on previous agreements, the consumer asks for a service and waits the producer to fulfill it, not taking into account how the request has been processed. So, the SP coordinates the activities of the involved OUs, ensuring their organizational autonomy.

SPs are characterized by pre- and post-conditions [28] that are, respectively, input and output parameters. In order to enable the Service Point, all the pre-conditions must be satisfied, while the post-conditions determine the SP success. In fact, it is a SP's task to coordinate the work activities monitoring the work during its execution, validating the executed activities and managing dynamic process changes and exceptions.

2.3.2. Service processes

In order to manage the coordination between partners, a SP uses the *Service Processes* (SPRs). To take part in cooperative projects, an organization must declare what it can offer to partners, using the description of the offered products and services. Moreover, business processes need feedback, in order to control their own work progress. However, business processes are part of the enterprise strategic core, as they represent the organization know-how and contain a lot of proprietary information. Since all the outcomes are the result of an internal business process, partners must not have direct access to this information.

A process service, conceptually based on the *Service Oriented Architecture* (SAO) paradigm, is an abstract representation of an enterprise private business process. The process service model is a layered model, where the process service layer is a public abstraction of the outcomes the enterprise is able to deliver, and the business process layer is the internal process flow in the enterprise.

Besides the outcomes (products and events) description, a process service definition (close to definition of [24]) contains the conditions surrounding the offers (inputs, guarantees, observable states, etc.), as well as the provider's information (access information, communications modes, etc.), and includes information dedicated to data and activities synchronization, transactional management, and retrieval of services state [23]. OUs offering the same service in different ways, publish two different SPRs.

Using the process service abstraction level, the cooperative process realization boils down to the problem of process services composition and integration. A cooperative process becomes a set of cooperative activities called SPs. These SPs coordinate the work of the partners involved and interactions between SPRs. To achieve this coordination, a SP implements several functions supporting computer-mediated cooperation for controlling, deciding and evolving the cross-organizational cooperative process.

2.4. Serviceflow modeling

The model described in this paper involves several actors belonging to different organizations. The aim is to create a communication channel in order to permit the exchange of information and documents. The design of such a system is a complex process because there are several requirements to consider. As stated by Wetzel and Klischewski [27] these requirements can be structured in four dimensions:

1. *Flexibility*. The process is characterized by an high variability so it has to support flexibility and adjustment of the services.
2. *Customer orientation*. The inter- and intra- patient's variation of needs has to be considered.
3. *Interoperability*. The software systems of the organizational units have to communicate with each other and to exchange knowledge and documents.
4. *Agreement*. All the actors involved in the management of the care process have to enter into agreements (the choice of IT support, the communication protocol and the timing and modalities of the service delivery).

In order to satisfy all these characteristics, we focused, in the design of the system, on *Serviceflow Management* (SM). This concept has been developed especially to meet the requirements of public service domains, but it can also be applied to other service domains such as the health-care.

While the client (the patient, in our case) naturally follows her/his individual concerns, the service provider (the VHCO) offers its performances as "pieces of work" (services) performed by one or more business organizations, based on the recognition and satisfaction of customer needs. The notion of *serviceflow* (SF) is meant to pick up both of these perspectives:

- From the customer's perspective, a serviceflow gives a customer the feeling of being embedded in a coherent "flow of service" taken care of by the service organization(s) where the service provided "follows", "accompanies" or "precedes" the customer as she/he moves through time and space.
- From the service provider's perspective, the emphasis is on the integration and coherence of all situated subservices across temporal, spatial and team boundaries, which are combined to form a continuous and complex overall service to satisfy the client's need (based on standard processes).

The notion SF indicates the interrelation of all subservices whereas what actually flows is

1. the customer's concern (which may evolve over time) related to a service agreement and his/her accumulated service experience (often supported by the customer's physical or virtual presence), as well as
2. the documented plan and history of each individual sequence of service tasks.

The aim of SM is to provide a process representation, which may serve as a basis for:

- cooperation agreements between the service providers involved in a SF (i.e., process patterns for standardized SFs);
- service agreements between client and customer (i.e., personalized process patterns);
- individual process documentation to be passed on between the service providers.

Also, modeling must allow for flexibility and decentralized control of what service tasks are to be carried out and what should be the schedule for service tasks to follow.

As mentioned before, we define SF in terms of SPs. We model the previously mentioned "pieces of work" as SPs, in order to obtain a SF as "The successive interrelation of a number of SPs" [27].

The advantage of this approach is twofold: from the patient's (customer) point of view, the coordination and synchronization of all the services needed is assured, while from the OUs (providers) point of view it permits to separate the definition of the internal workflow processes, focusing on the method of providing the services outside.

The model described requires, as stated by Weigand and van den Heuvel [29], commitments and agreements between the parties involved. These information must be formally represented into contracts, used by the SPs during the execution of the SF.

The interrelation of SPs is subject to possible changes: whereas the SF history (the sequence of SPs passed) is, of course, not changeable, the SF schedule, i.e., the part of the SF pattern with SPs not visited yet, may be manipulated by deleting or adding SPs or changing their order.

3. Technologies used to realize the serviceflow management system

3.1. Overview

The Serviceflow Management System is implemented by means of the following technologies:

- a *Contract*, which states the agreements between the parties involved, driving the choice of the services needed by the patient;
- an *Organizational ontology*, defining the organizational structure of the VHCO;
- *Workflow processes* implemented by means of the Workflow Management System YAWL [31].

Fig. 2 shows the implementation details of the System Architecture found in Fig. 1 in Section 2.2.

At the *coordination/communication* level, we realize the SF defining every SP as a task of a workflow. We refer to the binding of a SP and a task with a decomposition at the service level as Task/SP. The Task/SP activation is driven by the *contract*, which points out *What* services must be activated, *When* it is the moment to activate them, and *Who* are the Actors involved. The *organizational ontology* provides information about *Where* the Actors are located in the VHCO, and the role they have in the service.

Every Task/SP, when activated, starts a *sub-process*, located in the *service* level of the system architecture (see Fig. 1), which contains the definition of the SPRs provided by the involved OUs. These *sub-processes* are structured so that their execution is able to respect the interfaces defined for communication/cooperation purposes in the context of the service required.

At a communication level, since all the information exchanged through the SPs should be persistent and easily interpreted, we use XML documents. We exploit this language to formalize all the data and information exchanged, and also to define the Contract terms and the *organizational ontology*. The processes, from the one which implements the SF, to the SPRs and their interactions at the *service* level, are implemented using the open-source workflow management system YAWL.

In the remainder of this section, we elaborate on each of the technologies used in the design of our *Serviceflow Management System*.

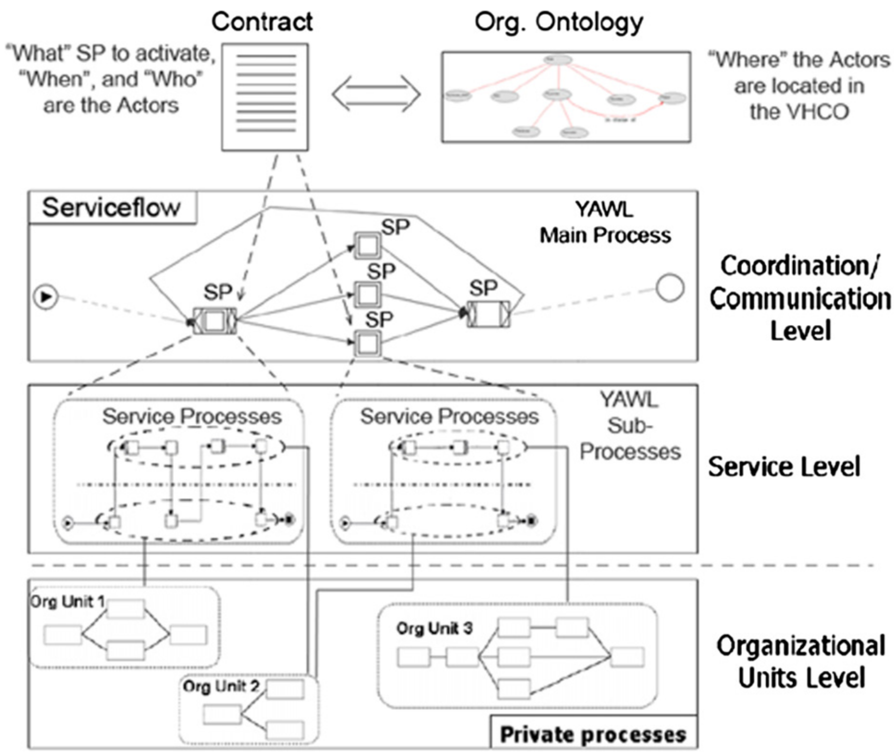

Fig. 2. Implementation details for the system architecture.

3.2. Organizational ontology

Research on ontologies is becoming increasingly widespread in the computer science community. An ontology is defined as "an explicit specification of a conceptualization" [8], and is now gaining a specific role in Artificial Intelligence and other fields, such as knowledge engineering [9–12], knowledge representation [13–15], medicine [16] and many others, including knowledge management and organization [13–15].

Ontologies allow to formalize a knowledge domain by means of concepts, attributes which characterize them, and relations between the concepts. Instances can be created to obtain real "objects" from the concepts defined. By means of these instruments it is possible to create a structured formalization of the domain which will be populated with real data by creating new instances. In this way, every instance is not just a data, but it has a meaning and is related to the other created instances.

The serviceflow management system uses an ontology in order to formalize an abstraction of the virtual health care organization involved in the patient's care process. This organizational ontology specifies the organizational units, the roles and the resources involved in the care process and their location. This provides the system with a general view of the organization in order to involve the correct agents and retrieve their needed information for every service activated.

Ontologies can be made machine readable by means of their XML representations. Many languages has been defined, such as RDF [17], or OIL [18], in order to translate the ontologies in electronic format, together with queries and restrictions on the data. We just need to represent the structure of the virtual health-care organization, so the XOL [19] language is chosen, because it can represent classes, attributes, relations and instances with great simplicity.

3.3. Contracts

An essential aspect of inter-organizational cooperation is the mutual commitments that parties must accept to integrate their processes [29]. Contracts are the most natural way to prescribe the coordination between the partners involved, and are used to make explicit the (legally binding) commitments the partners (agents) make, driving the actions and services to be performed. These commitments comprise the "glue" which integrates the autonomous organizational units into virtual "alliances", in order to formalize the shared goals and policies of the VHCO.

As mentioned in Section 2.3.1, SPs exploit the information contained in the contracts to monitor the interactions between partners and to activate the right services at the right time, involving the right people and OUs. To achieve this result, we state that contracts must be composed of four sections, at least:

- a *Legal* section defining the legal aspects of the contract;
- a *Who* section defining the agents involved (e.g., the patient and her/his physician);
- a *What* section describing contract subject and relationships to be satisfied (e.g., the type of Diabetes and the types of visits to be performed), and
- a *How* section defining the execution steps: activities to be performed, ways to perform them and their timings (e.g., frequencies of any of the visit defined).

Contracts must be machine readable, in order to be exploited by the system. In particular, they are formalized in XML language following the recommendations proposed by the Crossflow organization [5–7].

3.4. Yawl

In our serviceflow management system we use YAWL, an open source workflow management system. There are two reasons for using YAWL. First of all, YAWL is based on the well known workflow patterns [20], 2 and hence it is a more *expressive* language than any of the commercial systems on the market. Second, through the worklet service [46–48] YAWL allows for much more *flexibility* [38–44]. Both expressiveness and flexibility are clearly important in this domain. Therefore, we selected YAWL. In the remainder we briefly describe the basic concepts in YAWL and the ability to use Worklets.

In the last years, despite the efforts of the *Workflow Management Coalition* [1,2], the workflow products support languages that differ significantly in terms of concepts, constructs and their semantics [3,31]. Thanks to the workflow patterns initiative [33,32], the sets of the most relevant workflow patterns are now available, together with the documentation for integrating them in existing workflow languages [32,34,35]. A detailed description of these patterns can be found in [3,4].

To overcome these limitations and exploit the patterns defined, workflow researchers at Eindhoven University of Technology and Queensland University of Technology developed a language called *YAWL (Yet Another Workflow Language*) [36]. YAWL is inspired by Petri nets, but it is a completely new language with independent semantics. YAWL is highly expressive [31] and provides direct support for all the workflow patterns, has a formal semantics and offers graphical representations for many of its concepts.

The graphical representation of the language's modeling elements is shown in Fig. 3. A *workflow specification* in YAWL is a set of *extended workflow nets* (EWF nets) which form a hierarchical structure.

Tasks may be *atomic tasks* or *composite tasks* representing EWF nets at a lower level in the hierarchical structure. Each task can be instantiated multiple times using the concept of *multiple instances* for atomic or composite task. *Conditions* can be seen as places of a Petri net. Every process definition starts with a unique *input condition* and ends

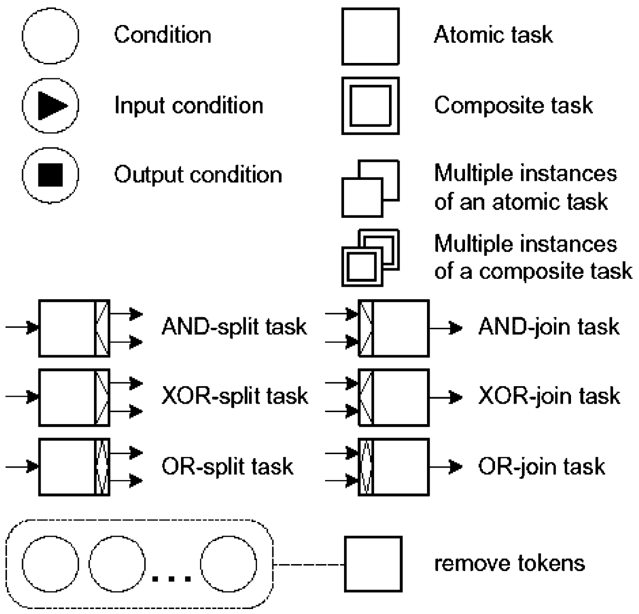

Fig. 3. Symbols used in YAWL.

with a unique *output condition*. *Connections* are possible between condition-task (and vice versa) and task-task. The latter may be interpreted as an implicit hidden condition between the tasks connected. *AND, XOR* and *OR* splits and joins are natively supported using the corresponding symbols in Fig. 3. YAWL also provides a notation for removing tokens from a specified region: the *remove tokens* construct.

This language is supported by the YAWL system, composed of the *YAWL Editor*, which permits the visual editing of workflow definitions and the data they exploit, and by the *YAWL Engine*, which performs the workflows execution.

An important point of extensibility of the YAWL system is its support for interconnecting external applications and services with the workflow execution engine using a service-oriented approach [31,46]. In particular, the *Worklet Dynamic Process Selection Service* for YAWL allows to substitute a workitem in a YAWL process with a dynamically selected *worklet*—a discrete YAWL process that acts as a sub-net for the workitem and so handles one specific task in a larger, composite process activity [49].

An extensible repertoire (or catalogue) of worklets is maintained. Each time the service is invoked for a workitem, a choice is made from the repertoire based on the data within the workitem, using a set of rules to determine the most appropriate substitution. The selected worklet is then launched as a separate case.

The service provides for dynamic ad-hoc change and process evolution, without having to resort to off-system intervention and/or system downtime, or modification of the original process specification. In particular, worklets are very useful to provide flexibility to this application, for example to enact different ways to perform a health-care service on the basis of the patient's data, or writing a set of rules to automatically select the right SPRs for every SP on the basis of the data stored in the contract. This feature is not yet exploited in our application, but we are working to manage the health-care process flexibility integrating the use of worklets in the process model.

This section presented all the technologies needed for the implementation of the *web-based health serviceflow management system*: the *Crossflow* project specifications to formalize the contract containing the main actors involved, disease treated, services provided with specified timings and constraints; the *XOL* language to integrate the organizational structure of the VHCO together with roles and resources involved in the care process; the *YAWL* process definition language for the implementation of the health-care process.

All these technologies are used for the implementation of the application, shown in the next section, while we are still working on *worklets* as a solution for the management of the process' flexibility.

4. Clinical application

4.1. Application scenario

We designed and built a *serviceflow management system* able to manage the interaction between the partners involved in the care process of the patient. It is guided by a formal contract which defines the patient's needs in terms of services in order to deliver the health service negotiated between health-service consumer (patient) and providers (health-care professionals). The system is able to manage the interactions between patient and health-care professional interactions through the *Web-based Health Serviceflow Management System* (WbHSMS).

The patient is asked to be periodically submitted to two kinds of visits in order to monitor the evolution of her/his health conditions: face-to-face encounters and virtual encounters. The system relies on an organizational ontology which provides a formal representation of the human agent roles and resources as well as material resources needed to manage the service she/he is willing to receive.

The main agents involved in this application are:

- *Patient*: the patient suffers from diabetes (even if the application could be adapted for almost any chronic disease). This kind of patient needs to be visited on a regular basis by her/his physician through face-to-face and virtual encounters and to access different OUs for some prescribed exams and tests. A health-care professional is committed to manage her/his therapy plan. The number and frequency of the encounters are defined in a formal contract that has to be subscribed during the first face-to-face encounter. All the care management activities must be performed in agreement with this contract.
- *Health-Care Professional* (HCP): she/he is involved in the care delivery of her/his patient, until the end of the therapy (if the patient dies or no other therapy is good for her/him). The HCP must assess the clinical relevance of the

health care information collected through the face-to-face and virtual encounters scheduled in the contract, and can execute the following actions: (1) to ask the patient for further information, needed for a better assessment of her/his health state; (2) to call the patient for an extra face-to-face encounter to quickly assess and solve an unexpected occurred problem; (3) to change the current therapy on the basis of the acquired information, and (4) to revise the contract terms, changing the face-to-face and/or the virtual encounters plan previously scheduled.

- *Web-based Health Serviceflow Management System* (WbHSMS): The user interface to the system is Web-based for the users accessing the Internet or based on the MAS interface. By means of the WbHSMS, the patient can send to her/his HCP the required personal information about her/his health status, life style and the results of clinical exams and/or tests performed in any organizational unit. Once contacted for a virtual encounter, the system can remind the patient the date of the next face-to-face encounter or read, through a Text-to Speech system, a message sent by the HCP. By the other hand, the HCP connects to the WbHSS to read the information sent by her/his patient, take the proper clinical actions on the basis of the information read and communicate these decisions to the patient.

All the interactions between the agents involved are formalized as "services" offered to the patient. The SF application and the data needed (contract, organizational ontology and data as XML documents) are realized by means of the technologies introduced in Section 3.

The components implemented and the clinical applications are presented in the next sections, following the structure of Section 3.

4.2. Organizational ontology for the application

In order to manage the interactions between the agents involved, this system must be able to know who the agents are, what are their rules, the resources available and where they are located. For example, the booking of a retinopathy screening could activate the HCP and an automatic booking system at the ophthalmologic lab, while the booking for a blood test could enact the communication between the patient and a human agent at the right health-care unit. The way a service is provided could change depending on the different actors involved. For this reason, we must formalize the whole VHCO in order to acknowledge the system about the organizational structure details. This organizational model is based on an ontology, which formally represent the organizational structure, the roles, the resources (human and material), involved into the SF and the relations between them.

The system exploits this ontology at the *coordination/communication* level of its architecture (see Figs. 1 and 2). Every time a SP is activated, the system uses the information stored in the ontology in order to enact the resources needed to provide the service.

The data in the ontology is updated "on the fly" in order to keep it consistent. For example, when a new physician registers into the system, her/his data are added as instance of the proper concept derived from the "Physician" one.

Fig. 4 shows a basic ontology which represents some of the most important resources involved in the care process. The resources formalized are human and technological. The human resources are composed by the physician (general practitioner or specialist), the patient and a technical operator. Technological resources comprise the MAS devices and the interfaces used for the virtual encounters.

Fig. 5 shows the part of the XML ontology defining the *Patient*'s concept, the *Physician*'s concept, the Patient's *Address* attribute and the relation *In charge of* which states that a *Physician* is "In charge of" a *Patient*.

4.3. Contract for the application

The agreements between HCP and patient about the therapy plans are formalized in a contract. This contract defines:

1. *Agents and disease treated*: it defines the patient's and the HCP's personal data (*Who* part of the contract). It also defines the disease treated (for example "Diabetes, Type II") as the *What* part of the contract.

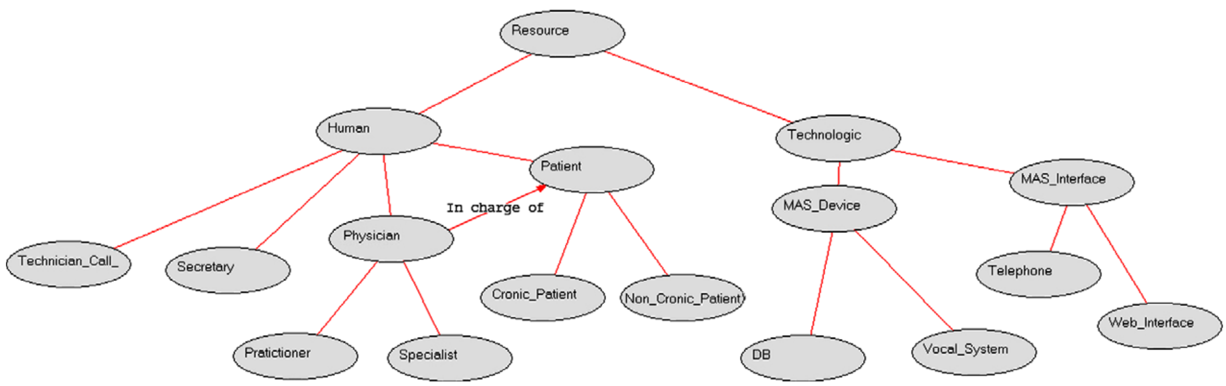

Fig. 4. Example of resources taxonomy for diabetes care.

```
<class>
  <name>Root</name> <documentation>Main Entity</documentation>
</class>

<class>
  <name>Resource</name> <documentation>Resource Class</documentation> <subclass-of>Root</subclass>
</class>

<class>
  <name>Human</name> <documentation>Human Resource</documentation> <subclass-of>Resource</subclass>
</class>

<class>
  <name>Patient</name> <documentation>Patient Resource</documentation> <subclass-of>Human</subclass>
</class>

<class>
  <name>Physician</name> <documentation>Physician Resource</documentation> <subclass-of>Human</subclass>
</class>

[...]

<!— Entities Attributes (in slots) -->

<slot>
  <name>Address</name> <documentation>Patient's Address</documentation>
  <domain>Patient</domain> <slot-value-type>string</slot-value-type>
</slot>

[...]

<!—Relations between Entities (in slots) -->

<slot>
<name>In charge of</name> <documentation></documentation>
<domain>Physician</domain> <slot-value-type>Patient</slot-value-type>
</slot>
```

Fig. 5. Piece of organizational ontology in XML language.

2. *Virtual encounters plan*: it defines what kind of information must be requested to the patient, and the frequency of the contacts between patient and WbHSMS to send this information to the HCP. For example, she/he must measure her/his blood pressure once a day, and communicate the values twice a week. On the other hand, the HCP must periodically read and analyze the data sent by the patient with a defined frequency.
3. *Face-to-face encounters plan*: it defines the period of time between two consecutive face-to-face encounters. Once an encounter has occurred, the patient must wait for this period to elapse before undergoing the next one. For this reason, the system must be able to remind the patient the date of the next encounter.
4. *Temporal constraints* for executing actions by HCP and patient according to the plans:
   - *HCP's temporal constraints*: she/he must analyze the data sent by the patient to the WbHSMS within a maximum time and must never ignore them. She/he must notify the completion of this operation every time she/he connects to the WbHSMS to read the data. She/he must also communicate to the patient the result of any face-to-face or virtual encounter within a prescribed maximum time.
   - *Patient's temporal constraints*: it is allowed that the patient undergoes a face-to-face or virtual encounter within a maximum time after the scheduled one.

The plans and the temporal constraints form the *How* part of the contract. As mentioned in Section 3.3, the contract is mapped in the XML representation of the *Crossflow project* [5–7] contract definition, and is updated on the fly any tme new information and data are provided during the execution of the application. The contract is exploited in the *coordination/communication* level of the system architecture (see Figs. 1 and 2), in order to activate the right services at the right time. For example, let the patient stipulate a contract for a rethinopathy screening twice a year, starting from the contract stipulation date. When the current day approaches the screening date, the *Face to Face Encounter Booking* service point is activated (see Fig. 9) in order to enable the contact between the patient (or her/his HCP) and the ophthalmologic lab and perform the booking of this face-to-face encounter.

4.4. Serviceflow application in YAWL

This is the main part of the application. Here, the relations between the agents involved become *services* available to the patient, and the overall care process is provided as the right sequence of services needed for her/him.

The application is implemented as a multi-level workflow: the main level is the implementation of the serviceflow (*Coordination/Communication* level), where every SP is realized as a task of the main level's process. The *Service Pro-*

cess level is implemented with sub-processes at the lower level of the workflow. The details of this implementation, together with examples that clarify the concepts, are presented in the following sections.

4.4.1. Coordination/communication level: the serviceflow model

The analysis of the interactions between the agents took to the identification of a set of *services*, formalized as *SPs*. The interrelation of those *SPs* generates the SF illustrated in Fig. 9. This serviceflow is the implementation of the model in the *Coordination/Communication* seen in Fig. 2.

The SF model also involves the definition of the data to be entered and managed, like personal information, clinical data to be examined, etc. Those data are formalized in XML language, to be easily integrated with the process definition. Even the contract and the organizational ontology are integrated as data in order to be easily exploited by the system.

Following the flow of services, we can see a first part of the ''contact'' between patient and HCP. First of all, the HCP registers to the system entering all her/his personal and professional data; then, the same does the patient also indicating who is the HCP responsible for him. Finally, a face-to-face encounter takes place, after a booking phase. During this encounter, the HCP assesses the patient's condition, decides the terms of the contract mentioned before and proposes them to the patient. Once HCP and patient agree on and subscribe the contract, its data is entered into the system in order to manage correctly the following services.

Fig. 6 shows the interface which allows the patient and her/his HCP to fill in the contract terms, at the end of the encounter. By means of this interface, it is possible to:

- confirm HCP's and patient's data (*who* section of the contract),
- the type of disease treated, the types of encounters (*what* section) and
- their timings (*how* section).

For example, this patient suffers from diabetes type I, so he stipulates a contract with his HCP in order to be followed during his care process. This contract states that the patient needs to undergo a retinopathy screening (a face-to-face encounter) twice a year, and to measure his glycemic level at home (as a virtual encounter) twice a day. These terms are then filled in by means of the user interface, and the system automatically updates the XML contract. Fig. 7 shows the piece of XML containing some of the patient's data, while the piece of XML contract containing the retinopathy screening and its timings is shown in Fig. 8.

After that, the flow splits in three main parts:

- On the top, we find the first phase of a virtual encounter: the first SP allow the patient to contact the WbHSS in order to send her/his health status data and receive potential recommendations and communications from her/his HCP. The next one allow the management of potential errors in the data sent.
- In the middle, the HCP concludes the virtual encounter by reading these data and taking actions on the basis of their clinical relevance: he can change something in the therapy or in the contract terms, or ask the patient for an urgent not scheduled face-to-face encounter. Changes and communications are entrusted to the WbHSS, which will propose them to the patient at the next contact for a virtual encounter.
- The bottom part allows the face-to-face encounters' execution: after a booking phase, patient and HCP meet for the encounter. After that, the HCP must communicate to the patient the encounter report and can also change something in the therapy or in the contract, or ask the

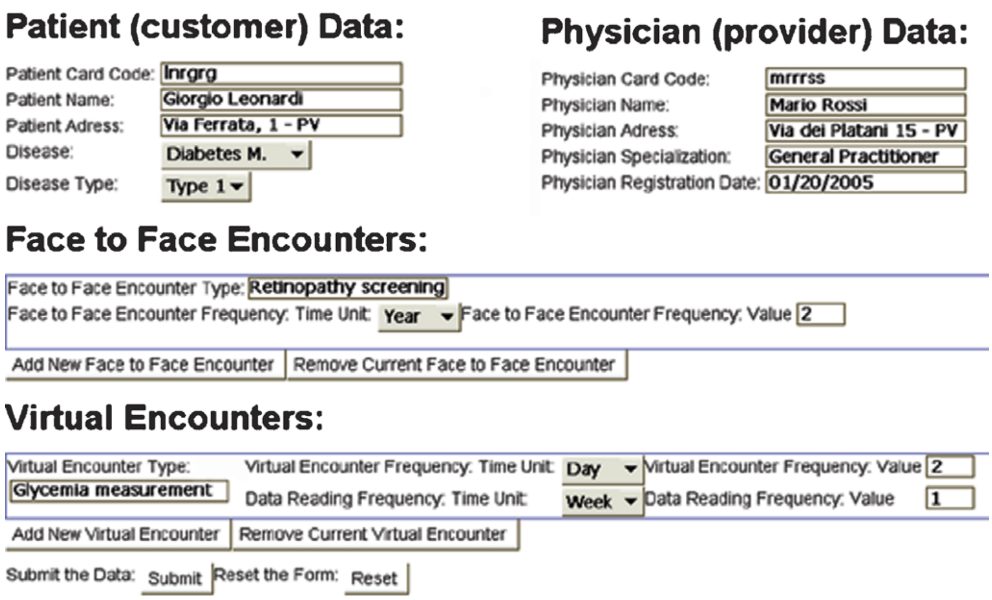

Fig. 6. Contract definition user interface.

```
<Contract>
    <DataSection>
        <Header>

        [...]

            <Consumer name="Patient">
            <Record>
                <RecElem field="CardNumber"><Simple type="string"><Value>lnrgrg</Value></Simple>
                </RecElem>

                <RecElem field="Name"><Simple type="string"><Value>Giorgio Leonardi</Value></Simple>
                </RecElem>

                <RecElem field="Address">
                <Record>
                    <RecElem field="Street"><Simple type="string"><Value>Via Ferrata</Value></Simple>
                    </RecElem>

                    <RecElem field="Number"><Simple type="integer"><Value>1</Value></Simple>
                    </RecElem>

                    <RecElem field="City"><Simple type="string"><Value>Pavia</Value></Simple>
                    </RecElem>

                    [...]

                </Record>
                </RecElem>

                [...]

            </Record>
            </Consumer>
```

Fig. 7. Patient's data mapped in the "Consumer" section of the contract.

```
<RecElem field="Visits">
    <ParamList>

    <ParamListElement index="0">
    <Record>
        <RecElem field="VisitType"><Simple type="string"><Value>Retinopathy Screening</Value></Simple>
        </RecElem>

        <RecElem field="VisitFreq">
        <Record>
            <RecElem field="TimeUnit"><Simple type="string"><Value>Year</Value></Simple>
            </RecElem>

            <RecElem field="Freq"><Simple type="integer"><Value>2</Value></Simple>
            </RecElem>
        </Record>
        </RecElem>

    </Record>
    </ParamListElement>
    </ParamList>
</RecElem>
```

Fig. 8. Retinopathy Screening (face-to-face encounter) data mapped in the "Parameters" section of the contract.

patient for a not scheduled face-to-face encounter. The communications can be done by voice, directly to the patient, or later, by means of the WbHSS.

The entire process terminates when no therapy gives any advantage to the patient, or in case of patient's death.

Every task represents a SP, so it can be activated only if the pre-conditions defined for the corresponding SP are satisfied (e.g., if it is time to undergo to an encounter) and can be considered executed only when the post-conditions are verified (e.g., if the encounter is concluded and the report has been compiled). To obtain this, the pre-conditions and the post-conditions are coded in the task definition and verified on the real data and on the contract terms.

4.4.2. Service process level: implementation of the service processes

When a Task/SP is activated, it is necessary to enact the right agents in the right OUs in order to fulfill the service represented by the SP. The enactment and the cooperation of the agents is managed by executing the right SPRs at the *Service Process* level. To obtain this, every Task/SP is decomposed in a sub-process (located at the *service level*

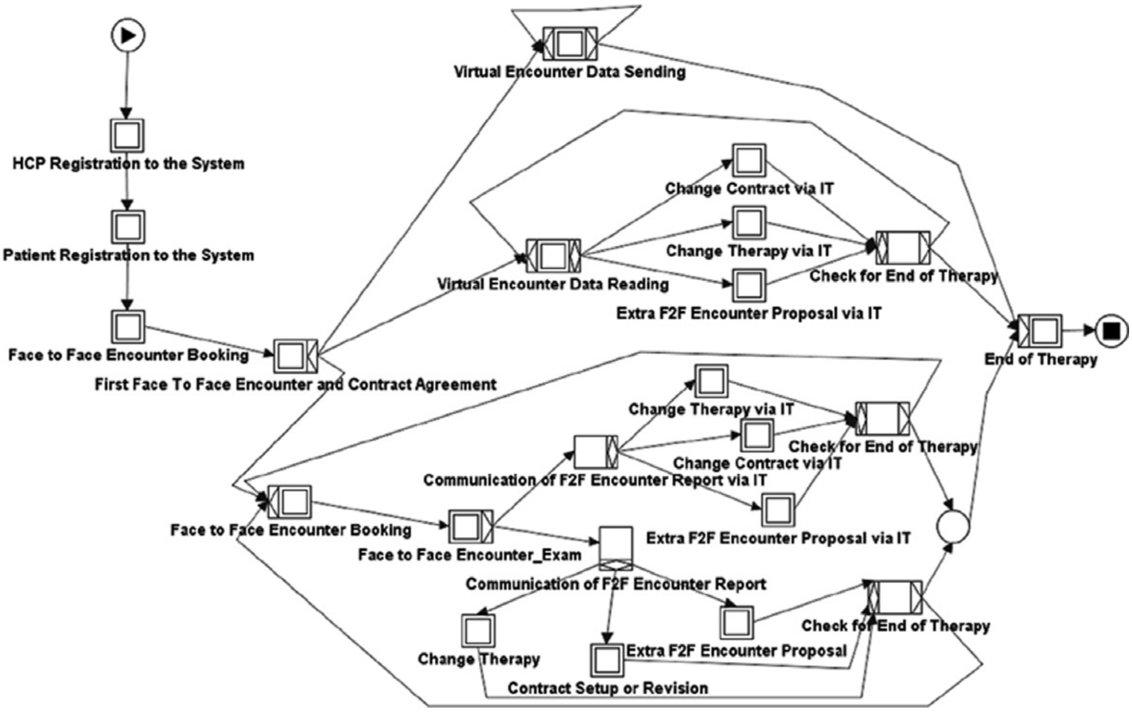

Fig. 9. Serviceflow model in YAWL language.

of the system architecture, see Fig. 2) that contains the definition of all the SPRs needed for the execution of the Task/SP. The operations defined in the SPRs are implemented as tasks of the sub-process, and these tasks are connected in order to coordinate the work of the involved OUs.

For example, the sub-process shown in Fig. 10 is called when the Task/SP "Face to Face Encounter Booking" is activated, and allows the booking of a face-to-face visit. It implements two *SPRs*: the one on the left defines the operations performed by the "Customer" (the patient or the physician who books the visit for him/her), while the

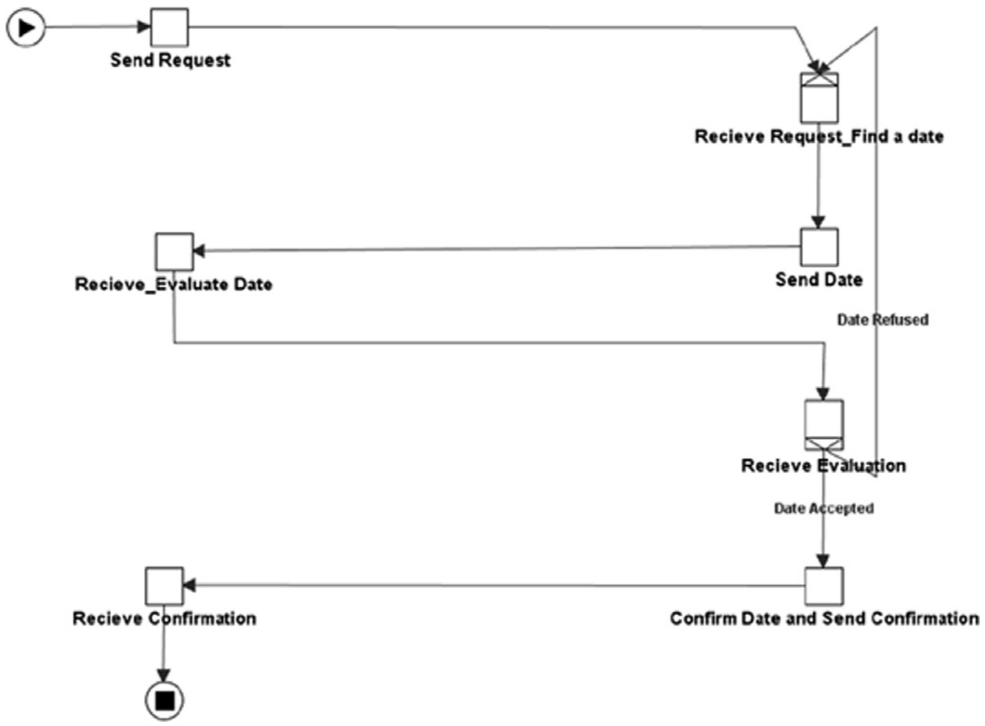

Fig. 10. Sub-process of the "Face to Face Encounter Booking" service point.

one on the right defines the operations performed by the "Provider" (the booking service of the organizational unit providing the encounter).

This sub-process starts with the *customer* sending the booking request for the encounter (*send request* task). The *provider* receives the request and searches for a suitable date (*receive request/search for date* task). Then, the *send date* task allows the *provider* to send a message containing the date found to the *customer*, who receives it and evaluates the date proposed (*receive/evaluate date* task). After the evaluation, the *customer* sends a message to the *provider*, to accept or refuse the date. The *provider* has two possibilities (*receive evaluation* task): if the date has been refused, he/it must return to the *receive request/search for date* task in order to find another date and propose it to the *customer* for a new evaluation; otherwise the booking is confirmed (*confirm date and send confirmation* task), and a confirmation message is sent to the *customer*. The sub-process ends when the confirmation is received (*receive confirmation* task).

The SF application is then implemented by means of a multi-level workflow. The main level implements the SF, mapping every *SP* and their pre- and post-conditions in tasks. The lower level implements the *SPRs* needed to execute every *SP*, grouping them in sub-processes called by the activated *SPs*. The contract contains information about what kind of services are needed and what are the timings to activate them. So, applying these timings to activate the right *SPs* at the right time, it is possible to generate the sequence of services representing the evolution of the care process for the patient. The *organizational ontology* is able to maintain roles, resources and the relations between them, in order to implement the *VHCO* structure, offering information about the agents involved for every *SP* to be activated.

The workflow model has been implemented in YAWL, which has been shown to be a valid instrument due to its high expressiveness and completeness. The application presented here is the first implementation of Serviceflow modeling in the medical field using YAWL as process definition language, as indicated in [37].

Currently we are investigating the use of YAWL in both a Dutch and a Chinese hospital in the context of the Brain-Bridge project. The goal is to focus on flexibility aspects (e.g., using worklets) and finding out how careflows are really executed (e.g., using process mining).

5. Conclusion

This paper presented an application of *Serviceflow* in the diabetes care field. As many chronic diseases, diabetes treatment requires the patient to move to different *organizational units* to meet the right agents, in order to receive the health services she/he needs and to communicate information about her/his health status and results of tests and visits.

In order to meet these requirements, we focus on *Serviceflow Management* (SM), that allows to model the health-care process on the basis of *"services"* tailored on the patient's needs. In particular, these services are offered by a single *Virtual Health-Care Organization* (VHCO) that contains the definition of the entire organizational structure in order to hide its complexity to the patient. The VHCO provides to the patient *face-to-face* and *virtual* encounters through a *Web-based Health Serviceflow Management System* (WbHSMS), in order to follow her/him also at home.

The WbHSMS is implemented using the *YAWL* language, and the process is guided by a *formal contract* that defines: (1) the actors involved in the care process; (2) the type of disease treated; (3) the type of encounters needed by the patient and (4) the timings of their execution. The application described in this paper is the first one in medical field implemented with the YAWL language.

The information in the contract allows the WbHSMS to activate the right services at the right time, while the *organizational ontology* defining the VHCO provides information about the actors and/or the agents to enact when a service is activated.

The process model can be adapted to different chronic pathologies and home-based care, by simple changes in the contract or in the model if necessary.

Future work will involve the following activities: (1) write custom services for YAWL in order to export the process data and interface the application to a health-care oriented database; (2) implementing and deploying the system in real-life environments; (3) use worklets [49,46–48] to add flexibility to the process model, and (4) extending the approach with process mining [53,51,52] capabilities to analyze and improve running care processes.

References

[1] Lawrence P, editor. Workflow handbook 1997, Workflow management coalition. New York: Wiley; 1997.
[2] Fisher L, editor. Workflow handbook 2001, Workflow management coalition, future strategies. FL: Lighthouse Point; 2001.
[3] van der Aalst WMP, ter Hofstede AHM, Kiepus-zewsky B, Barros AP. Workflow patterns. Distrib Parallel Database 2003;14(1):5–51.
[4] Workflow Patterns Home Page. Available from: http://www.workflowpatterns.com.
[5] Ludvig H, Whittingham K. Virtual enterprise co-ordinator-agreement-driven gateways for cross-organizational workflow management. In: Proceedings of the international joint conference on work activities coordination and collaboration, San Francisco, February 1999, p. 29–38.
[6] Vonk J, Derks W, Grefen P, Koetsier M. Cross-organizational transaction support for virtual enterprises. In: Etzion O, Scheuermann P, editors. Cooperative Information Systems (COOPIS 2000). Berlin: Springer; 2000. p. 323–34.
[7] Crossflow Home Page. Available from: http://www.crossflow.org.
[8] Gruber TR. A translation approach to portable ontology specification. Knowl Acquis 1993;5:199–220.
[9] Gruber TR. A translation approach to portable ontology specifications. Knowl Acquis 1996;5:199–220.
[10] Uschold M, Gruninger M. Ontologies: principles, methods and applications. Knowl Eng Rev 1997;11(2):93–136.
[11] Gaines B. Editorial: using explicit ontologies in knowledge-based system development. Int J Hum Compt Syst 1997;46:181.

[12] Gomez-Perez AK. Knowledge sharing and reuse. In: Liebowitz J, editor. Handb Exp Syst. CRC Press; 1997.
[13] Guarino N. Formal ontology, conceptual analysis and knowledge representation. Int J Hum Comput Stud 1995;43(5/6):625–40.
[14] Artale A, Franconi E, Guarino N, Pazzi L. Part-whole relations in object-centered systems: an overview. Data Knowl Eng 1996;20(3):347–83.
[15] Sowa JF. Knowledge representation: Logical, philosophical, and computational foundations. Brooks/Cole Publishing Co., 1999, ISBN: 0-534-94965-7.
[16] Gangemi A, Pisanelli D, Steve G. Ontology alignment: experiences with medical terminologies. In: Guarino N. editor. Formal ontology in information systems, IOS Press (this volume), 1998.
[17] Lassila O, Swick R. Resource description framework (RDF) model and syntax specification, W3C proposed recommendation, January 1999. Available from: http://www.w3.org/TR/PR-rdf-syntax.
[18] Horrocks I, Fensel D, Harmelen F, Decker S, Erdmann M, Klein M. OIL in a Nutshell, knowledge acquisition, modeling and management, 2000, p. 1–16.
[19] Karp R, Chaudhri V, Thomere J. XOL: an XML-based ontology exchange language, Tech Rep, July 1999.
[20] Holman H, Lorig K. Patient self-management: a key to effectiveness and efficiency in care of chronic disease, Public Health Reports, May–June, 2004.
[21] Wagner EH, Austin BT, Von Korff M. Organizing care for patients with chronic illness. Millbank Q 1996;74:511–44.
[22] Edworthy SM. How important is patient self-management? Best Pract Res Clin Rheumatol 2000;14(4):705–14.
[23] Perrin O, Godart C. A model to support collaborative work in virtual enterprises. Data Knowl Eng 2004;50(1):63–86.
[24] Schuster H, Baker D, Cichocki A, Georgakopoulos D, Rusinkiewicz M. The collaboration management infrastructure. In: Proceedings of the 16th international conference on data engineering (ICDE), February 2000. San Diego, CA: IEEE Computer Society; 2000. p. 677–8.
[25] Giorgino T, Azzimi I, Rognoni C, Quaglini S, Stefanelli M, Gretter R, et al. Automated spoken dialog system for hypertensive patient home management. Int J Med Inform 2005;74(2–4):159–67.
[26] Bellazzi R et al. Design, methods, and evaluation directions of a multi-access service for the management of diabetes mellitus patients. Diabetes Technol Ther 2003;5(4):621–9.
[27] Wetzel I, Klischewski R. Serviceflow beyond workflow? Concepts and Architectures for Supporting Inter-Organizational Service Processes, Advanced Information Systems Engineering. In: Proceedings 14th CAiSE, Springer lecture notes in computer science, Berlin; 2002. p. 500–15.
[28] Klischewski R, Wetzel I, Baharami A. Modeling serviceflow. Information systems technology and its applications. In: Proceedings ISTA 2001. Bonn: German Informatics Society; 2001. p. 261–72.
[29] Weigand H, van den Heuvel WJ. Cross-organizational workflow integration using contracts. Decis Support Syst 2002;33(3):247–65.
[30] Panzarasa S, Maddè S, Quaglini S, Pistarini C, Stefanelli M. Evidence-based careflow management systems: the case of post-stroke rehabilitation. J Biomed Inform 2002;35(2):123–39.
[31] van der Aalst WMP, Aldred L, Dumas M, ter Hofstede AHM. Design and implementation of the YAWL system. In: Proc. of the 16th Int. Conf. on advanced information systems engineering (CAiSE 04), Riga, Latvia, June 2004.
[32] van der Aalst WMP, ter Hofstede AHM, Kiepuszewski B, Barros AP. Workflow Patterns. Distrib Parallel Database 2003;14(1):5–51.
[33] Workflow Patterns Home Page. Available from: http://workflowpatterns.com.
[34] Riehle D, Zúllighoven H. Understanding and using patterns in software development. Theor Pract Obj Syst 1996;2(1):3–13.
[35] van der Aalst WMP, ter Hofstede AHM. YAWL: yet another workflow language. Inform Syst 2005;30(4):245–75.
[36] YAWL Home Page. Available from: http://www.citi.qut.edu.au/yawl.
[37] YAWL Impacts Page. Available from: http://www.citi.qut.edu. au/members/research_areas/bpm/projects/yawl/impacts.jsp.
[38] van der Aalst WMP, Jablonski S. Dealing with workflow change: identification of issues and solutions. Int J Comput Syst Sci Eng 2000;15(5):267–76.
[39] Agostini A, De Michelis G. Improving flexibility of workflow management systems. In: van der Aalst WMP, Desel J, Oberweis A, editors. Business process management: models, techniques, and empirical studies, vol. 1806. LNCS, Springer; 2000. p. 218–34.
[40] Casati F, Ceri S, Pernici B, Pozzi G. Workflow evolution. In: Proceedings of ER '96, Cottubus, Germany, October 1996. p. 438–55.
[41] Ellis CA, Keddara K. A workflow change is a workflow. In: van der Aalst WMP, Desel J, Oberweis A, editors. Business process management: models, techniques, and empirical studies, vol. 1806. LNCS, Springer; 2000. p. 201–17.
[42] Herrmann T, Hoffmann M, Loser KU, Moysich K. Semistructured models are surprisingly useful for user-centered design. In: De Michelis G, Giboin A, Karsenty L, Dieng R, editors. Designing Cooperative Systems Coop 2000. Amsterdam: IOS Press; 2000. p. 159–74.
[43] Klein M, Dellarocas C, Bernstein A, editors. In: Proceedings of the CSCW-98 workshop towards adaptive workflow systems, Seattle, WA, November 1998.
[44] Klein M, Dellarocas C, Bernstein A. Adaptive workflow systems. J Comput Support Coop Work 2000;9:3–4. Special issue.
[45] Schmidt K, Simone C. Coordination mechanisms: towards a conceptual foundation of CSCW systems design. Computer supported cooperative work: The Journal of Collaborative Computing, vol. 5. Netherlands: Kluwer Academic Publishers; 1996. p. 155–200.
[46] Adams M, ter Hofstede AHM, Edmond D, van der Aalst WMP. Facilitating flexibility and dynamic exception handling in workflows. In: Belo O, Eder J, Pastor O, Falcao e Cunha J, editors. Proceedings of the CAiSE'05 forum, FEUP, Porto, Portugal; 2005. p. 45–50.
[47] Staffware, Staffware Process Suite Version 2. White Paper, Staffware PLC, Maidenhead, UK, 2003.
[48] Weske M. Formal foundation and conceptual design of dynamic adaptations in a workflow management system. In: Proceedings of the thirty-fourth annual Hawaii international conference on system science (HICSS-34), Sprague, R, Los Alamitos, CA: IEEE Computer Society Press; 2001.
[49] Adams M, ter Hofstede AHM, Edmond D, van der Aalst WMP. Worklets: a service-oriented implementation of dynamic flexibility in workflows. In: Meersman R, Tari Z, et al., editors. On the move to meaningful Internet systems 2006, OTM confederated international conferences, 14th international conference on cooperative information systems (CoopIS 2006), vol. 4275. LNCS, Springer, 2006. p. 291–308.
[50] Harrison S, Dourish P. P, Re-Place-ing space: the roles of place and space in collaborative systems. In: Proceedings CSCW 1996. p. 67–76.
[51] van der Aalst WMP. Process Mining in CSCW systems. In: Shen W, James A, et al., editors. Proceedings of the 9th IEEE international conference on computer supported cooperative work in design (CSCWD 2005). Coventry University/IEEE Computer Society Press; 2005. p. 1–8.
[52] van der Aalst WMP, Alves de Medeiros AK, Weijters AJMM. Genetic process mining. In: Ciardo G, Darondeau P, editors. Applications and theory of Petri nets 2005, vol. 3536. LNCS, Springer; 2005. p. 48–69.
[53] van der Aalst WMP. Discovering coordination patterns using process mining. In: Bocchi L, Ciancarini P, editors. First international workshop on Coordination and Petri Nets (PNC 2004), STAR, Servizio Tipografico Area della Ricerca, CNR Pisa, Italy, 2004. p. 49–64.