Domain Knowledge-based Automatic Workflow Generation*

Soon Ae Chun, Vijayalakshmi Atluri, and Nabil R. Adam

CIMIC and MS/IS Department
Rutgers University
Newark, New Jersey, USA
{soon,atluri,adam}@cimic.rutgers.edu

Abstract. The traditional workflow design paradigm relies heavily on humans who statically specify business processes. However, such a manual design approach is not suitable for many cases: (a) *Inter-agency workflows* that cross autonomous organizational boundaries require experts who possess knowledge required for defining workflows composed of services from the constituent organizations; (b) *Customized workflows* that require many variations make it infeasible and error-prone to predefine the complex workflow in advance. This paper presents an alternative design paradigm that allows automatic on-the-fly generation of a workflow schema. This *dynamically integrated workflow composition approach* utilizes conceptual ontologies of domain services (tasks) and domain integration knowledge that serves as a model for workflow integration (composition) rules. It also uses user profiles (preferences) as part of (transient) compositional rule base. We present a prototype system for "registering a new business" in the State of New Jersey that implements this automatic workflow design methodology.

1 Introduction

Workflow management systems (WFMS) have made it possible to automate the coordinated execution of different business processing activities. WFMS take over the responsibility of controlling the coordinated execution of tasks from human coordinators. This automation is possible with well-formed workflow specifications. Currently, tasks and interdependency conditions among tasks need to be fully specified by humans, before the workflow is executed. Various modeling techniques have been proposed [5, 7], but the modeling process remains manual. Graphical tools [14] can be helpful, however, the graphical objects still need to be put together manually. The workflow design process typically requires to specify all the possible use cases and different execution environments. Once the workflow has been designed, modification of the workflow definition is costly and time consuming. Manual workflow specification needs automation, given the

* This work is partially supported by the National Science Foundation under grant EIA-9983468 and by MERI (the Meadowlands Environmental Research Institute).

fact that workflow technology is widely being adopted for web-based B2B and B2C e-commerce transactions as well as eGovernment services in G2G, G2B and G2C.

Often, government services span agency boundaries e.g., for new business registrations [1], welfare and social services [4], etc. The workflow for these services is referred to as *inter-agency workflow*. An inter-agency workflow is composed of component services provided by autonomous government agencies. The decentralized execution model proposed in [3] allows automatic execution with the collaborative coordination of participating agencies' services and information systems, while maintaining autonomy of each agency. In this paper, we emphasize that inter-agency workflow design needs to be dynamic and automatic rather than static and manual.

Our approach is based on a knowledge-based workflow design method. An ontology of services serves as a domain model for component services (tasks) and their relationships. An ontology of government regulations serves as a domain model for compositional rules. We present a *dynamic workflow model* that makes use of the tasks and rules in the ontology hierarchies, and of user profiles. Compositional rules consist of selection rules that select obligatory and preference tasks, and coordination rules that glue tasks together in order. Each rule is represented as Condition-Action pair. The composition algorithm evaluates compositional rules against a user profile, automatically generating the customized inter-agency workflow. This knowledge-based approach allows automating the complex inter-agency workflow design as well as provides a customized workflow that requires fewer evaluations at run time.

Problem: In the following, we provide two example scenarios to illustrate the need for a dynamic and automatic workflow design method.

Example 1. *A developer, Bill, wishes to build a warehouse complex on his property on the bank of the Hackensack River in Little Ferry, NJ. His property of 15 acres (block 108.04, lot 2.04) is currently vacant and designated as "Light Industrial and Distribution B" zone. His property is only two feet below the maximum tide level. He will need to fill it with some material to secure a sound foundation.*

A workflow designer needs to know that (1) Bill needs to acquire a zoning certificate from the NJMC; (2) if the parcel falls into the so-called environmentally sensitive area, such as wetlands, riparian land, and flood plains, etc., Bill may need permits from local, state and federal agencies, including NJ Department of Environmental Protection (NJDEP) Waterfront Development Permit, Stream Encroachment Permit, Water Quality Certificate, Army Corps of Engineers Section 10 or Section 404 permit, Riparian Grant, etc.

The designer needs knowledge on local, state, and federal regulations in order to properly design a workflow for Bill. In addition, the designer needs to consider not only Bill's case, but many others with different geo-spatial characteristics. Specifying all the possible variations for every user in one workflow is not feasible. The next example illustrates the need for variant workflows for "busines registration process."

Example 2. *John Smith would like to start a new auto body repair shop in New Jersey. Jane Carlson would like to open a convenience store. John Smith wants to locate his business in downtown Newark, Essex county. Jane Carlson wants to have her shop near her home in Mercer county. John wants to have the business incorporated with several employees, while Jane wants to have it in sole proprietorship without any employees.*

According to these requirements, two different inter-agency service specifications are needed as shown in figures (1a) and (1b) for John and Jane, respectively. While both entrepreneurs need to file the tax-related business registration application at the Division of Taxation, their required steps and tasks differ. John will also need a certificate of incorporation and employee related insurance as well as an air quality permit. Our automatic workflow design methodology ben-

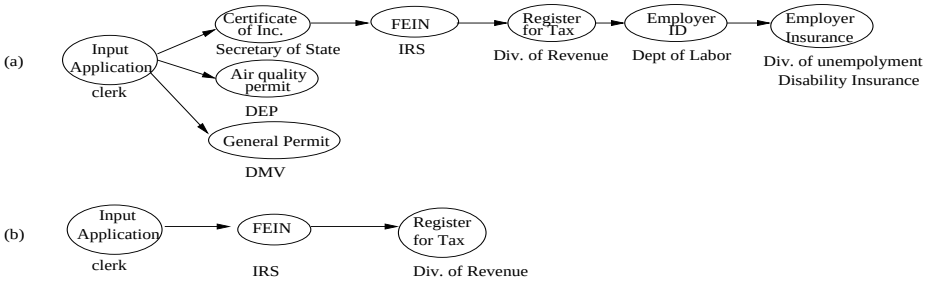

Fig. 1. Customized business registration process

efits the designer, and also citizens who without the automated design system need to find which regulations by which agency apply for their particular situations. This is time-consuming and complex process, as different regulations are enforced by different autonomous agencies. Moreover, there are few people with comprehensive knowledge of the regulations and services of different agencies.

In Section 2, we present the domain knowledge-based workflow generation model Section 3 presents an algorithm to generate the workflow definition and describes our prototype system. Section 4 discuss related work, followed by conclusions and future research in Section 5.

2 Automatic Workflow Composition Model

2.1 Task and Service Ontology

For service discovery and composition, services are described as service interfaces for external use and internal implementations. We argue that descriptions of individual services do not capture the inter-relationships among services. The linear set of individual services without any structures can result in inefficiency in service discovery and identification. For this reason, we use an ontology of services. An ontology is defined as a description of concepts and relationships that exist among them [8]. For example, the knowledge of a composite service *Open a New Business* is represented as a *component hierarchy* (Figure 2). Each node

represents a service, and each down link (up link) represents a *has-component* (*component-of*) relationship. The *has-component* relationship emanating at a service, s, defines all the component services of s. The knowledge of tasks and their

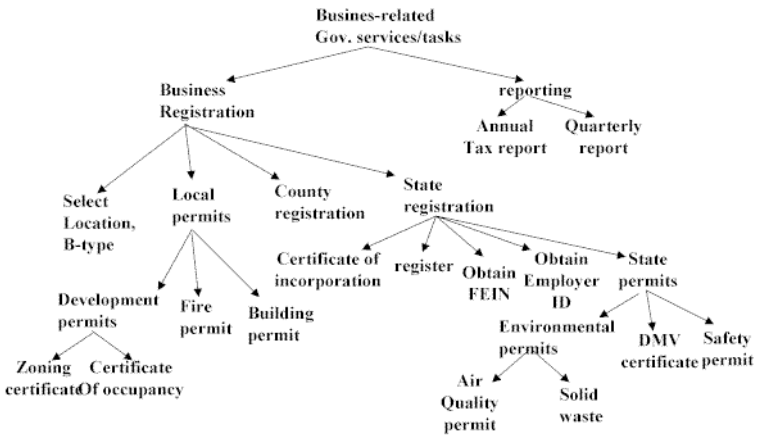

Fig. 2. Component Hierarchy for a Service of opening a new business

relationships explicitly considered in our workflow generation model facilitates the search for appropriate component tasks via *component-of* relationships and provides necessary interface and connection information needed for composing integrated services.

Definition 1. [**Domain Services/Tasks Ontology**] *A domain service ontology SO is defined as a set of services* $S = \{s_1, s_2, ...\}$ *where each* $s_i \in S$ *is defined as a triple* $\langle id, SA, SR \rangle$*, where id is a service identifier, SA is a set of service attributes and SR a set of relationships that are associated with* s_i*. The service attribute SA is represented as a 5-tuple* $s_i = \langle a, Pre, A, Input, Output \rangle$*, where a denotes the execution agent, or service provider of* s_i*, Pre denotes the precondition set for* s_i*, A the set of activities (or operations) within* s_i*, Input the set of input parameters to* s_i*, and Output the set of output parameters of* s_i*. SR is a set of relationships* $\{rel_1, rel_2, ...\}$ *where each* rel_i *is a triple* $\langle name, N, mode \rangle$ *with name for relationship name, N for a set of services that bear* rel_i *relationship with* s_i *and mode to denote whether relationship* rel_i *is obligatory or optional with tasks linked in the service ontology.*

We use the notation $rel_i(s_i)$ to retrieve all the services that bear the relationship rel_i to s_i, i.e. N. A relationship $SR = \{$*has-component component-of, implement* $\}$ for s_i defines a set of all component services for s_i, a set of services that s_i is a component of, and a set of regulatory rules that s_i enforces, respectively.

2.2 Ontology of Government Regulations

Government services implement federal, state and local regulations [12, 9]. Obvious examples include: "a drivers license is required for vehicle operation," "low

income households are entitled to social benefits," etc. However, some regulations are not so obvious. Moreover, as seen in examples in 1, regulatory knowledge required for designing an inter-agency workflow crosses the boundaries of local, state, and federal agencies. Regulatory rules may include different types:

1. Semantic Rules: These are the rules that affect the activities involved in a workflow. For example, an environmental protection regulation states that any business type, such as an autobody shop, that releases a certain level of spray paint into the air is required to obtain an air quality permit.
2. Spatial Rules: These rules are concerned with the geographic features of a business [10], e.g. (1) If a development project includes construction of a structure along, in, or across the stream channel, or 100 year flood plain of any stream, then a stream encroachment permit is required. (2) If a business is located in municipalities under a certain agency jurisdiction, then a development application needs to be filed and approved by that agency.
3. Temporal Rules: Temporal rules may state the absolute and relative deadlines for a task. For instance, if a public records filing for a new business entity is submitted, then tax registration forms must be submitted within 60 days.
4. Sequencing Rules: These rules specify the obligatory sequences among tasks. For example, obtaining a certificate of incorporation is required before business registration.

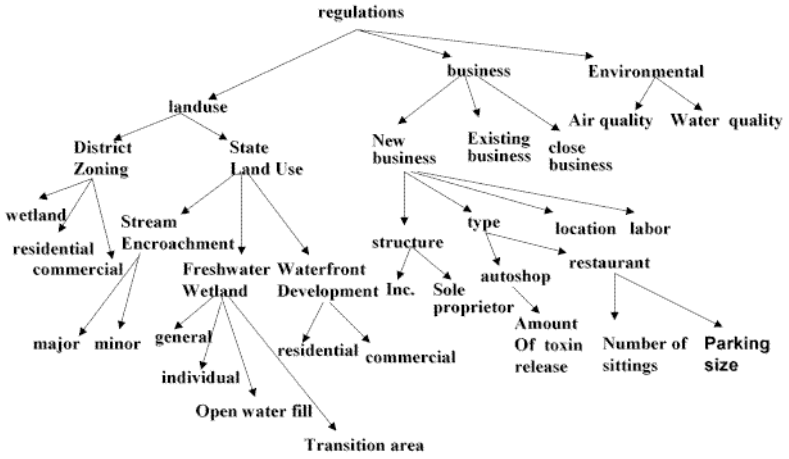

Fig. 3. Government Regulations Ontology

The comprehensive knowledge about regulations is conceptually modeled as regulatory topic ontology (Figure 3). This topic-subtopic hierarchy allows the identification of necessary information (e.g. type, location and kind of a new business) and regulation information. Each leaf node is associated with a regulatory rule.

The rule nodes are modeled as Condition-Action relations (Figure 4), where Condition is a logical expression and Action is an operation on services (e.g. insert, parallel, order). This rule node has a link to the text document containing

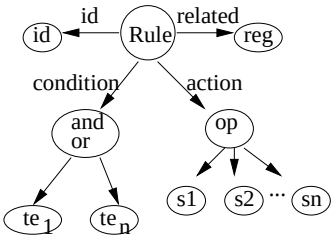

Fig. 4. Model of a compositional rule

the regulation. The rule node also has a link to actual services that implement it (i.e. an *implemented-by* relationship). For example, the leaf node on the subtopic of *incorporated* has a relationship *is-subject-to* pointing to a regulatory rule *reg*. The rule *reg* is associated with the text document containing regulations such as "If the business structure is incorporation, then it is required to obtain a certificate of incorporation." It also has a relationship of *implemented-by* to a service "Obtain a certificate of incorporation" in the service hierarchy described above.

The following definitions formalize the regulations ontology and composition rules.

Definition 2. [**Domain Rules Ontology**] *Domain rules ontology RO is defined as a set of rule topics* $O = \{o_1, o_2, ...\}$. *Each* $o \in O$ *is defined as a 4 tuple* $< id, RA, Rel, R >$, *where id denotes an rule topic identifier, RA denotes a set of attributes, Rel a set of relationships that c bears, and R is a set of domain rules,* $\{r_1, r_2, ...\}$ *where each* $r_i \in R$ *is defined as a triple of* $(id, body, reg)$ *where id denotes the rule identifier and body is denoted as Condition-Action pair* $\langle c, a \rangle$ *where c and a are expressions for the antecedent and consequent of the rule, respectively. The regulation that is related to the rule is denoted by reg, which has a set of attributes* $\{rat_1, rat_2, ...\}$ *and a set of rule relationships* $\{rrel_1, rrel_2, ...\}$.

The condition c is defined as a logical expression, called trigger expression, as follows:

Definition 3. [**Trigger Expression**] *Given an alphanumeric string variable* $v \in \mathcal{S}$, *a function* $f \in \mathcal{F}$ *which is a set of predefined functions, a relational operation* $op \in OP = \{<, , =, \neq, \not<, \not>, \geq, \leq, \not\geq, \not\leq, \in, \subset, \subseteq, \notin, \not\subset, \not\subseteq, \supset, \supseteq, \not\supset, \not\supseteq\}$ *and a literal* $l \in \{\mathcal{S}, \mathcal{N}\}$ *for strings* $\mathcal{S}$ *and numbers* $\mathcal{N}$, *the* trigger expression *te is defined as follows:*

1. $(v\ op\ l)$ *is a trigger expression*
2. $(f(\boldsymbol{x})\ op\ l)$ *is a trigger expression*
3. *If* te_1 *and* te_2 *are trigger expressions, then* $\neg te_1$, $te_1 \vee te_2$, *and* $te_1 \wedge te_2$, *are also trigger expressions.*

The function f represents predefined semantic, spatial and temporal functions such as *amount(x), distance(x, y), after(x, y), etc.* An example trigger expression is $(number(employee) \geq 1)$. The action $a \in r_i$ is an operation $o \in \{insert, order, parallelize, ...\}$, that can be applied to tasks T in a workflow. Following shows semantics of actions: $insert(t_1)$ is an action to insert task t_1; $order(t_1, t_2)$ imposes an order: t_1 before t_2; $parallelize(t_1, t_2)$ allows t_1 and t_2 to be executed in parallel. The following is an example of a rule defined above:
(R1, $\langle$ *amount(spray-paint)* $\geq$ *1/2 gallon per hour*, *insert(obtain air-quality-permit)* $\rangle$, REG_1)
$R1$ states that if the amount of spray paint used is more than half a gallon per hour, then insert task *obtain air quality permit* into the workflow and add the trigger expression *amount(spray-paint)* $\geq$ *1/2 gallon per hour* as the precondition for the task. This rule is related to regulation REG_1.

2.3 User Profile Model

While the service ontology and regulatory ontology are static most of the time, the *user profile* varies from case to case. Our approach contains a user profile for customization as an important part of the model. The individual user profile is modeled as a goal and a set of preference attributes and their values. The user goal in the model identifies a domain (i.e., a set of services) for a composite service, and user preferences are used for composition rule evaluation to select component services. User preferences include functional preferences (e.g. the type of business), geographic preferences (e.g. business location), or temporal preferences (e.g. business opening date), and others.

The preference attributes required for composing a workflow are also derived from the regulatory topic hierarchy. For example, the business registration related user profile attributes include business type, location, kind and labor information.

Definition 4. [**User Profile**] *A user profile P is defined as a pair of $\langle g, PR \rangle$ where g is a user's goal service $g \in S$ that a user intends to initiate, and PR is a set of service preferences $\{p_1, p_2..., p_n\}$. Each p_i is represented as a pair of an attribute and its value, $\langle at, v \rangle$.*
The following is a profile for John Smith:

- P_{John} =$\langle$ *open new business*, { $\langle$ *type, in corporated* $\rangle$, $\langle$ *name, Car Care* $\rangle$, $\langle$ em kind, Autobody shop $\rangle$, $\langle$ *location, 80 Mercer Street, Newark, NJ 07052* $\rangle$, $\langle$ *employee_number*, $\geq$ *3* $\rangle$ } $\rangle$

2.4 Workflow Composition Function

Definition 5. [**Workflow Composition Functions**] *Given S, a set of services in service ontology SO, R a set of rules in rule ontology RO and P a set of user preferences, A rule selection function σ is defined as $\sigma : (S \times R \times P) \rightarrow CR$, where $CR = \{r_1, r_2, ..\} \subseteq R$ where each r_i is a pair $\langle c, a \rangle$ for condition c and*

action. Given CR from σ, workflow composition functions h and k denote task selection function and task coordination function, respectively, and are defined as follows:
1. $h : CR \rightarrow T$, h is a function that selects a set of tasks $T \in a(CR)$ where $a(CR)$ is an action a of each rule $r \in CR$. for each $t \in T$, and $Pre(t) = te$ is precondition value is set to a trigger expression $te \in c(CR)$
2. $k : CR \rightarrow D$, k maps all tasks $T \in CR$ into dependency set $D = T \times T$ according to the coordination action $a \in CR$.

Definition 6. **[Workflow]** *A workflow W is defined as a directed graph $G = (T, D)$ where $T \subseteq S$ is a node set representing a set of tasks selected by composition function h and $D \subseteq (T \times T)$ is an edge set for coordination dependencies among T, generated by composition function k.*

Intertask dependencies D support a variety of workflow coordination requirements. Basic types of task dependencies include *control-flow dependencies* which specify the flow of control based on the outcome state of a task (e.g. begin-on-success), *value-dependencies* which specify the control flow based on the output value of a task (e.g. number of employee ≥ 2), and *external dependencies* which specify the control flow based on certain conditions satisfied by parameters external to the workflow (e.g. task t starts at 9:00am) [2, 13, 3].

3 Dynamic Workflow Composition

This section presents components and an algorithm to dynamically generating a customized workflow.

3.1 Profile Gathering

User profile attributes are derived from the rule ontology and hierarchically organized. The profile attribute data required for registering a new business shown in Figure 5 consist of attributes like structure, location, name, type and employee, and so on. A leaf node represents an attribute that assumes a value. For instance, the attribute *structure*, can take a value from the set {*incorporated, sole proprietorship, limited partnership, ...*}.

This hierarchical profile organization is represented as a set of profile rewrite rules (Table 1). These rules are used to gather profile information by dynamically expanding the LHS of a rule with RHS nodes and collapsing RHS nodes when all the attributes are filled with values. Figure 6 illustrates the expansion and shrinkage of the tree structure as the profile information is gathered.

3.2 Automatic Integration of Customized Workflows

Given a set of customization rules and a user profile, the customization process generates an individualized workflow, selecting only relevant component services (i.e. tasks) and obeying other constraints such as sequence ordering rules. Given the customization rule base in Tables 2, and user profile, the customization process automatically selects tasks and arranges these tasks according to the

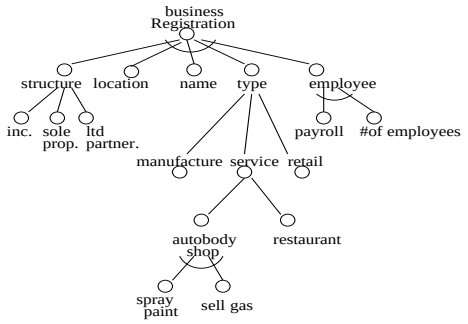

Fig. 5. Profile Attribute Hierarchy

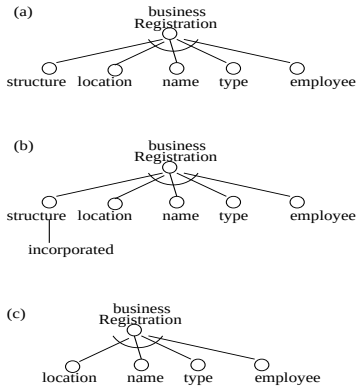

Fig. 6. Tree Structure for Profile Evaluation

Table 1. Profile Rewrite Rules

P1: $business\ registration \rightarrow structure \wedge location \wedge name \wedge type \wedge employee$
P2: $structure \rightarrow incorporated$
P3: $structure \rightarrow sole\ proprietorship$
P4: $location \rightarrow street \wedge county \wedge state \wedge zip$
P5: $type \rightarrow manufacture$
P6: $type \rightarrow service$
P7: $service \rightarrow autobody\ repair$
P8: $autobody\ repair \rightarrow amount(spray\ paint) \geq 1/2\ gallonhr \wedge sell\ gas$
P9: $employee \rightarrow payroll \geq \$1000 \wedge number(employee) \geq 1$

coordination rules. For instance, when a user enters *incorporated* as his business structure, the system finds rule R1 in Table 2, and inserts t_1 and t_2 into the workflow. Here, we assume t_1= *register business name with the State of NJ*, t_2 = *file original business certificate*, and t_3 = *register business's name with the County Clerk*. If the user prefers to establish a business of *sole proprietorship*, then R5 fires. As a result, t_3 is inserted into the workflow. As soon as a rule fires, the profile evaluation tree shrinks, and the next profile attribute evaluation starts (Figure 6). This process repeats until all the attributes of the profile tree acquire values and are used to fire appropriate rules. The end result of this process is a workflow. In addition to the tasks selected based on preferences, there exist

Table 2. Service Selection and Coordination Rules

R1: $\langle struct = incorporated, insert(t1) \wedge insert(t2) \rangle$
R2: $\langle struct = limited\ liabilitycompany, insert(t1) \wedge insert(t2) \rangle$
R3: $\langle struct = limited\ partnership, insert(t1) \wedge insert(t2) \rangle$
R5: $\langle struct = sole\ proprietorship, insert(t3) \rangle$
R7: $\langle register\ business, register\ business\ for\ tax \rangle$
R8: $\langle register\ business, obtain\ federal\ employer\ id \rangle$
R9: $\langle register\ business, certificate\ of\ incorporation \prec register\ business\ for\ tax \rangle$
R10: $\langle register\ business, fein \prec register\ business\ for\ tax \rangle$

obligatory tasks required for all users. For instance, *register-business-for-tax* is required for all applicants. These obligatory tasks are inserted by rules such as

R7 and R8 in Table 2. All the tasks in the workflow are then subjected to the inter-task sequencing rules to ensure the coordination requirements, such as R9 and R10 in Table 2. For instance, rule R9 states that if the process is *register business*, then the task *obtain a certificate of incorporation* should be finished before *register business for tax*.

The following algorithm generates an inter-agency workflow given a user's request for a service g. It uses the ontologies defined in 2.1 and 2.2.

Algorithm 1.[Generate Customized workflow]
Input: g: user's desired service
Output: W=(T, D) W=workflow with set of tasks T and dependency D
BEGIN
$T = \emptyset$, $D = \emptyset$
$S =$*has-component(g)* /* identify all component services for G */
$R = \emptyset$ /* set of rules applicable to services in S */
For each $s_i \in S$ { /* identify regulations and rules related to S */
    $reg = implement(s_i)$
    $r =$ *related-to(reg)*
    $R = R \cup r$ }
For each $r = (c, a) \in S$ { /* add obligatory service in workflow tasks T */
    if $(c(r) = \emptyset$ {
        $T = T \cup t \in a(r)$ }
For each $t_i \in T$ and $r = (c, a) \in R$ { /* add dependencies among tasks */
    if $a(r) = order(t_i, t_j)$ {
        $D = D \cup (t_i, t_j)$
        $Pre(t_j) = bs$
        $R = R - r$ }
S = S -T
PR = gather-profile*(g) /* gather relevant user profile */
While(PR $\neq$ empty) Do { /* while preference list is not empty */
    Given $p \in PR$
    For each $r = (c, a) \in R$ { /* evaluate preference rules */
        if $(eval(c, p) = true)$ {
            if $a = insert(t_i)$ { /* add a task into workflow */
                $T = T \cup t_i$
                $Pre(t_i) = te \in c$ } /* add trigger expression as precondition for t_i */
                For each $t_i \in T$ { /* add dependency */
                    if $a = order(t_i, t_j)$ {
                        $D = D \cup (t_i, t_j)$
                        $Pre(t_j) = Pre(t_j) \wedge bs$
                        $R = R - r$ }}}}
    $PR = PR - p$
    $S = S - t_i$ }
W = (T, D) }
END
*We assume the function *gather-profile* returns a set of attribute and value sets required for a service g.

The algorithm first identifies all the candidate component tasks based on the user requested service, g, as specified in the component service ontology. Second, it identifies applicable regulatory rules that are used to select obligatory and other tasks and determine dependencies among tasks. Selected tasks are composed according to the order related rules. The output of this algorithm is the customized composite workflow $W = (T, D)$ for the requested service g.

3.3 Prototype System

We have developed a prototype system[1] of generating customized workflows for the NJ State government [1]. Specifically, we have built an inter-agency service,

[1] http://cimic.rutgers.edu/dgov/demos1.html

called "a new business registration" process that requires information and coordination of various State agencies including Department of Revenue, Division of Commercial Recording, Division of Community Affairs, Department of Environmental Protection, Department of Public Health and Safety and others. The prototype system gathers the user profile information through an interview session, which is guided by interview rules which are based on case analysts' expert knowledge, i.e. ontology, such that only questions relevant to a specific business of the user are asked. The user profile gathered at the interview session and the business related regulations database are matched to generate a customized workflow. Web-based client applications for entrepreneurs include (1) a GIS-based business location service; (2) an interview interface; and (3) a workflow interface.

4 Related Work

Industry standards and research efforts such as WSDL, UDDI, ebXML, DAML-S, describing the semantics and metadata of web services, facilitate discovery and access of services, but they go only half way toward automating integration of services. The automation of service integration requires the selection and control knowledge that specifies what services are needed in certain circumstances. This knowledge of service composition and selection is left to domain experts to define manually, or for program developers to combine with the application code. Our approach of utilizing a domain model provides the capability to automate the integration and customization of workflow design.

Platforms for developing composite e-services proposed in eFlow [6], E-speak [11], and CMI (Collaboration Management Infrastructure) [15] are compositional models, similar to our approach. However, their design of e-services (compositional process) is still manual, while our approach uses a workflow definition model where the workflow task selection and ordering knowledge is explicitly modeled for automating the generation of the workflow definition.

5 Conclusions and Future Work

We have presented a formal model for *automatic workflow generation* that uses *domain knowledge* represented as service ontology, regulatory ontology and a user profile. We have presented the composition algorithm and data structures used for automatic generation of the workflow. This framework can be easily adopted to process definitions in other application domains where process design needs to obey company policies and other constraints. Our future work includes semi-automatically constructing the ontology of regulations and extracting business rules from documents. We also plan to address the workflow analysis and verification which provide validation of the automatically generated workflow. The dynamic modifications of workflow definitions to run-time changes in regulations, user profiles, or services are also under investigation.

References

1. N. Adam, F. Artigas, V. Atluri, S. Chun, S. Colbert, M. Degeratu, A. Ebeid, V. Hatzivassiloglou, R. Holowczak, O. Marcopolus, P. Mazzoleni, and W. Rayner. E-government: Human centered systems for business services. In *The First National Conference on Digital Government*, pages 48–55, May 2001.
2. Nabil R. Adam, Vijayalakshmi Atluri, and Wei-Kuang Huang. Modeling and Analysis of Workflows using Petri nets. *Journal of Intelligent Information Systems, Special Issue on Workflow and Process Management,*, 10(2), March 1998.
3. V. Alturi, S. Chun, and Pietro Mazzoleni. A Chinese Wall Secuity Model for Decentralized Workflow Systems. In *Eighth ACM Conference on Computer and Communications Security (CCS-8)*, Philadelphia, Pennsylvania, USA, November 2001.
4. A. Bouguettaya, A. Elmagarmid, B. Madjahed, and M. Quzzani. A web-based architecture for government databases and services. In *The First National Conference on Digital Government*, pages 56–59, May 2001.
5. F. Casati, S. Ceri, B. Pernici, and G. Pozzi. Conceptual Modeling of Workflows. In *Proceedings of the 1995 OOER Conference*, December 1995.
6. Fabio Casati, Ski Ilnicki, and Li-Jie Jin. eflow: a platform for developing and managing composite eservices. Technical report, HP Laboratories Palo Alto, March 2000.
7. Dimitrios Georgakopoulos, Mark Hornick, and Amit Sheth. An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. *Distributed and Parallel Databases*, pages 119–153, 1995.
8. T.R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. In N. Guarino and R. Poli, editors, *Proceedings of International Workshop on Formal Ontology*, Padova, Italy, 1993.
9. C. S. Han. *Computer Models and Methods for a Disabled Access Analysis Design Enviroment*. PhD thesis, Department of Civil and Environmental Engineering, Stanford University, June 2000.
10. R. Holowczak, S. Chun, F. Artigas, and V. Atluri. Customized geospatial workflows for e-government services. In *ACM-GIS 2001, The Ninth ACM International Symposium on Advances in Geographic Information Systems*, Atlanta, GA, USA, November 2001.
11. Alan Karp. E-speak e-xplained. Technical report, HP Laboratories Palo Alto, July 2000.
12. Shawn Kerrigan, Gloria Lau, Liang Zhou, Gio Wiederhold, and Kincho H. Law. Information infrastructure for regulation management and compliance checking. In *The First National Conference on Digital Government*, pages 167–170, May 2001.
13. Marek Rusinkiewicz and Amit Sheth. Specification and Execution of Transactional Workflows. In W. Kim, editor, *Modern Database Systems: The Object Model, Interoperability, and Beyond*. Addison-Wesley, 1994.
14. Asim Sadiq and Mria Orlowska. Applying a generic conceptual workflow modeling technique to document workflows. In *The second Australian Document Computing Symposium*, Melbourne, Australia, April 1997.
15. H. Schuster, D. Georgakopoulos, A. Cichocki, and D. Baker. Modeling and composing service-based and reference process-based multi-enterprise processes. In *Proceedings of the Inernational Conference on Advanced Information Systems Engineering (CAiSE)*, Stockholem, Sweden, June 2000.