WorkFlow Analyzed for Security and Privacy in using Databases

Wouter Teepe[1] Reind van de Riet[2]

Martin Olivier[3]

1) Department of Technical Cognition Science,

Rijksuniversiteit Groningen,

Groningen, the Netherlands

e-mail: wouter@teepe.com

2) Department of Mathematics and Computer Science

Vrije Universiteit

Amsterdam, the Netherlands

e-mail: vdriet@cs.vu.nl

3) Department of Mathematics and Computer Science

Rand University

Johannesburg, South Africa

e-mail: molivier@rkw.rau.ac.za

June 26, 2000

Abstract

When companies interchange information about individuals, privacy is at stake. On the basis of the purpose of the information interchange, rules can be designed for an agent (Alter-ego) to determine whether the requested information can be provided. This purpose can be derived from a WorkFlow specification according to which employees (agents) of one company are executing their tasks. Direct information flow as well as information which might flow through private and covert channels is considered.

Keywords: Security & Privacy and Database systems, Workflow, Cyberspace, Object-Oriented Databases.

1 Introduction

In a study, being conducted for many years, we have introduced the notion of Alter-ego, which is an object/agent, representing people in Cyberspace and acting on behalf of these people (see [vdRB96b]). We are in particular interested in the problem of protecting someone's privacy.

In choosing a company for certain services, like an insurance, an individual will not only look at conditions such as financial costs and benefits, but also at less tangible conditions such as privacy rules held in that company. Only after inspection of these rules a client may decide to accept an offer.

In the near future we expect that the privacy conditions are treated even more seriously, in that the client demands that an agent is placed in that company which checks the behaviour of the employees in that company. This agent may inspect WorkFlows according to which these employees are working. Actually, the situation can be even more complicated by considering the cooperation of this company (A) with another company (B), who is interested in information about the individual. The agent may then decide whether to give that information or not. Also in this case the agent may inspect the WorkFlow according to which employees in company B are working, in order to determine whether it is of the interest of the individual or contrary to his/hers privacy concerns.

The main point of this paper is therefore to analyze a WorkFlow specification to find out properties relevant for privacy protection. A simple example will make clear what kind of properties we mean. Suppose the company is an insurance company and the individual wants to be sure that decisions within this company about policies and about claims are made without taking into account specific properties of the individual, such as the colour of the skin or marital and social status. Using the tools described in this paper that individual can carry out this analysis.

It is assumed that two companies A and B have organized the work of all their employees in the form of a WorkFlow (abbreviated in the following as WF). Although WF systems have been in existence for a number of years, the trend towards greater interconnection will greatly impact such systems. On the one hand, interaction will involve more and more non-human participants. On the other hand the participants in WF processes will become more and more unrelated. The key to secure implementation of future generation WF systems is proper authentication and authorization of participants in a WF process. It is our contention that Alter-egos (see the next section) are particularly suitable for authentication, while roles are particularly suitable for authorization. We have presented these ideas in another paper [GRBO97].

These WFs are open for inspection. From them one can determine whether employees have the proper information to perform their tasks, but also whether they may have too much information about an individual, or even when they can conspire/cooperate with other employees and then

are able to derive private information about the individual, using special channels for information flow. For the agent or Alter-ego in company A it is possible to analyze these WFs. The outcome can be used in two different ways:

1. To determine whether a company is trustworthy with respect to keeping privacy rules the employees have (just) enough information about the individual, and
2. to decide whether to respond to a query which is sent to company A by one of B's employees taking into account the information this employee already has about the individual as derived from the WF; this is depicted in figure 1.

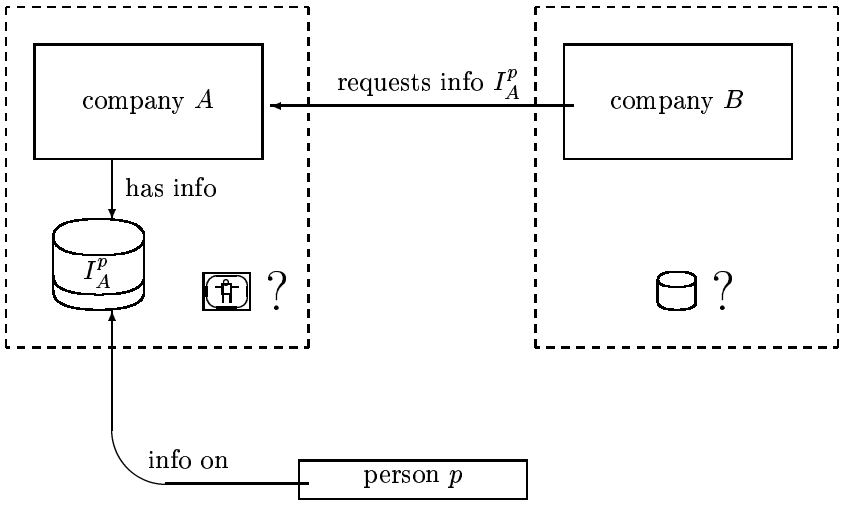

Figure 1: Company A keeping information about person P; Company B asking information about P. " ? " and " ? " indicate the possible existence of an agent that protects data and a database, respectively.

In the next section we will sketch a bit of the framework in which we are doing our research, i.c. the notion of Alter-ego and introduce the COLOR-X system which we use to specify WF diagrams. In the following section we then shall see how the COLOR-X diagrams can be represented as a Prolog database, so that analysis programs can be written in Prolog. The purpose is to analyze beforehand what each agent in B knows about P; that is to say: what s/he has to know in order to do the task specified, and what s/he may know when using information sent by cooperating/conspiring colleagues, flowing through a private and covert channels.

In the next section we will see how the privacy rules may use the knowledge provided in the analysis of the WF diagram. Finally, we will give some conclusions and hints for future work.

2 Background

2.1 Alter-egos

For our research in Security & Privacy in Cyberspace, we assume that individuals, either in an office environment, or in their homes, will be represented in Cyberspace by objects, called Alter-egos, in the sense of Object-Oriented Technology. The identifier of this object may be considered a combination of Social Security Number and e-mail address. The contents of these objects represent the properties of the persons for which they are Alter-ego; their behaviour can be seen as the behaviour of agents acting on behalf of these persons. They were introduced in [GRBO97], where it was shown how these Alter-egos can be structured and how Security and Privacy (S&P) aspects can be dealt with questions around responsibility and obligations of Alter-egos have been discussed in [vdRB96b, vdRB96a].

The main idea was to use Alter-egos to provide high-level security as discussed in [GRBO97], was that if the underlying communication system of Cyberspace ensures that every message contains an unforgeable Alter-ego of the sender (or initiator), one can design more powerful and higher level protection mechanisms than those existing today and which rely mainly on encrypting messages. In another paper [RJOS00] we describe an extension of the Mokum system which can cope with the situation as sketched here where different companies and systems are involved. The Mokum system is an object-oriented system in which objects can be defined communicating with each other by sending messages, just like people in an organization. In that paper we show how S&P rules can be securely maintained, "securely" meaning: provably secure. We also show how Alter-egos can be implemented in a distributed way, where parts are residing in databases of different companies.

2.2 The WorkFlow system COLOR-X

In this section we briefly describe the COLOR-X system in which it is possible to specify a WF diagram. In Workflow management (WFM) applications there are tasks to be completed by some organization, but the organization procedures require that this task will be carried out in steps where each step is executed by a different individual and no step can be performed before the steps it depends on are completed [GHS95]. We shall demonstrate a certain WFM-tool, COLOR-X, developed by our group to model Information and Communication Systems, using linguistic knowledge, and we will see how S&P rules can be derived from COLOR-X diagrams.

WFM tools are currently being used to specify how people and information systems are co-operating within one organization. There are at least three reasons why WFM techniques are also useful in Cyberspace. First, organizations tend to become multi-national and communication takes place in a global manner. Secondly, more and more commerce is being done electronically.

This implies that procedures have to be designed to specify the behaviour of the participants. These procedures may be somewhat different from ordinary WFM designs, where the emphasis is on carrying out certain tasks by the users, while in commerce procedures are based on negotiating, promises, commitments and deliveries of goods and money. However, as we will see, these notions are also present in the WFM tool we will use. Thirdly, people will be participants in all kinds of formalized procedures, such as tax paying or home banking.

2.3 Workflow and Security

This being said, how can we derive security and privacy rules from the Workflow diagrams (WFDs)? Specifying tasks and actions of people working in an organization naturally also involves the specification of their responsibilities [vdRB96b, vdRB96a, GRBO97]. This is what WFDs usually do. Responsibility implies access to databases to perform certain actions on data of individuals.

A Workflow Authorization Model is proposed in [AH96]. Authorization Templates are associated with each Workflow task and used to grant rights to subjects only when they require the rights to perform tasks. A Petri net implementation model is also given.

3 The Insurance-claim Application

The following example is about the treatment of a claim within an Insurance Company IC, concerning a trip booked with a Travel Agent TA. First we describe the processes in natural language, using numbers identifying the actions for easy identification with the boxes used in the COLOR-X diagram, following next.

There is an Insurance Company, IC, Furthermore there are persons, which can be employees of the IC. An employee can be an approver, a travel agent, an expert or a cashier. Also a person can be a submitter of a claim. The static (incomplete) structures are depicted in figure 2.

```
type person is_a thing                type employee is_a person
  has_a name                            has_a salary
  has_a address                         has_a function
type approver is_a employee
type expert is_a employee             type cashier is_a employee
type submitter is_a person            type travel_agent is_a employee
```

Figure 2: The static structure of the participants of the game.

Next follows the text of the Claim example. The numbers refer to the numbered boxes in the diagram.

1. A submitter SU sends in a triple of data (the trip TR, the incident IN, the amount AM1) to the approver AP of the insurance company IC.

2. AP receives the message from SU and creates an object called claim, CL, from the triple sent and asks the travel agent TA to verify the claim (possibly) within one week.

3. TA tries to verify CL within one week and return the answer to AP; if he is not able to do that :

4. Upon not receiving an answer from TA, AP assumes the claim is not OK and informs the submitter SU accordingly. (in a more realistic setting AP would send a reminder to TA)

   Upon receiving an answer from TA, which is "not OK", AP informs SU that the claim is not OK.

5. When TA's answer is positive and the amount is smaller than $100, AP asks the cashier CA to transfer the money to SU's Bank and informs SU accordingly.

6. Upon receiving an answer from TA, which is "OK", and the amount not being smaller than $100, AP asks an expert EX to look at the claim and AP informs SU appropriately.

7. EX treats CL and reports the decision to AP, which, in case the claim is found "not OK", handles as above in 4;

8. when "OK", AP determines the amount AM2 to be paid to SU and asks the cashier CA to transfer the money to SU's Bank;

9. CA pays the amount AM2 to SU's Bank.

We now give some clarification of the COLOR-X specification in figure 3.

- each box of actions has a mode: PERMIT, NEC or MUST. MUST means an obligation based on some negotiating in the past: as we are not sure that the action is actually carried out within the prescribed time it is necessary to define a counter measure indicated by the lightning arrow. The mode NEC means we can be sure the action is necessarily carried out by the system. PERMIT means there are no pre-conditions: the actions in the box can be executed;

- the actions are described in a formal language involving the participants and their roles.

It is important to notice that in sending a message from an agent A to a receiver R there are parameters in the message referring to objects, only the object identifiers are readable by R, not the contents. Only when the Workflow specifies that R has to do something with the contents

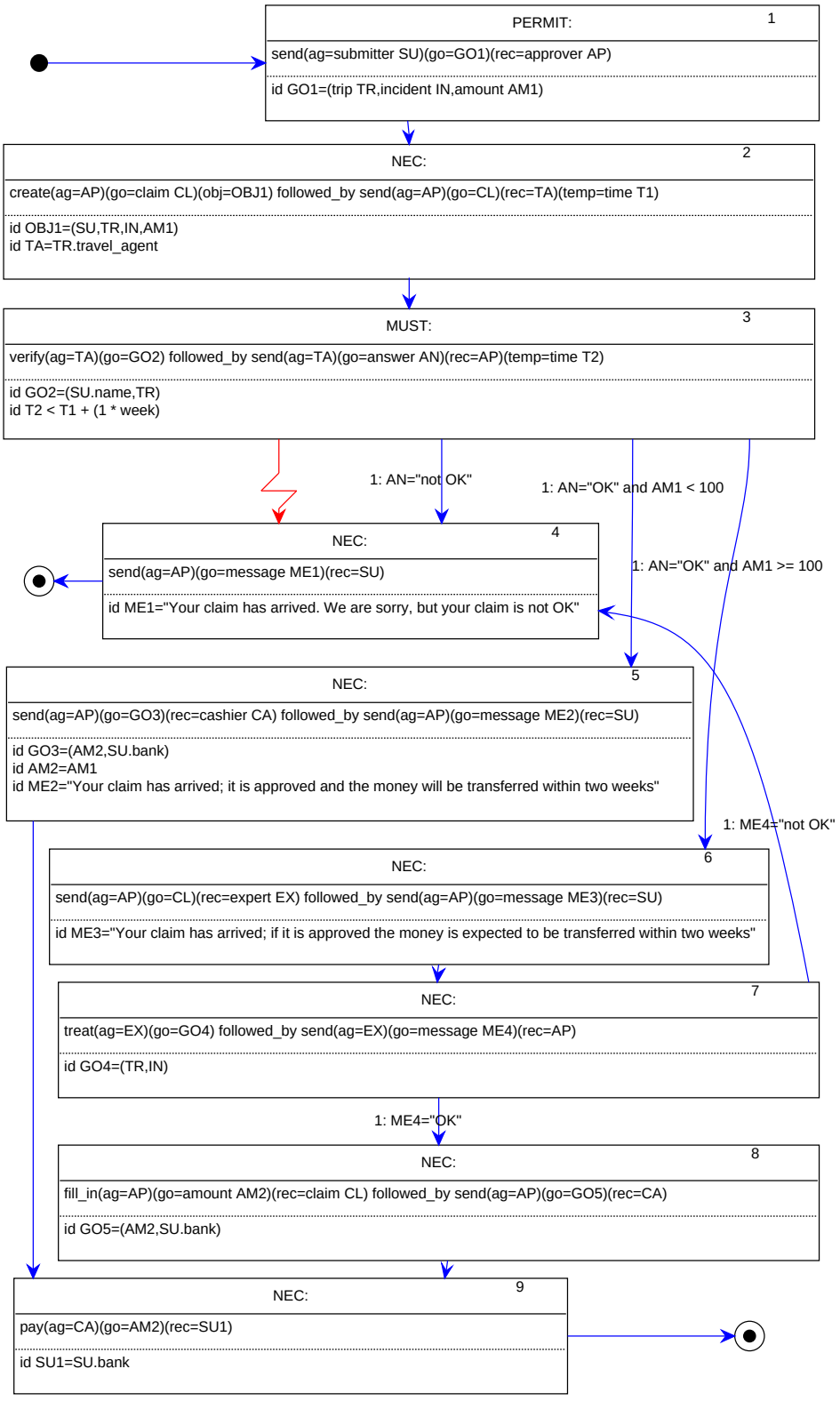

Figure 3: The claim example.

of the object, it is allowed to read this. This principle is also maintained in the Mokum system where security is based on hiding content, i.e. attribute values, instead of keeping secret object identifiers and, in general, names of files, procedures, triggers and attributes. The reason is clear: names of files can easily be given away. Before the UNIX system was developed, protection was generally based on the notion of "security by obscurity", so passwords were kept secret on a file, the name of which was known only to the system administrators. UNIX introduced the notion of one-way functions encrypting the passwords and making the password file publicly available.

4 The representation of COLOR-X diagrams in Prolog

Before stating for what purpose the WorkFlow WF is going to be analyzed we give three examples:

1. Suppose an agent like the travel agent TA asks information to company A, say the marital status of the submitter SU. The submitter's agent in A could refuse to respond because the analysis of WF shows that TA knows the trip TR and SU's name. The agent may reason as follows: TA knows TR, which was a trip for married people, if TA also knows that SU is single, TA may sue SU for bringing with him/her a partner to which s/he was not married.

2. Now the travel agent asks for the submitter's age. A more subtle way of reasoning could be the following: Although it is not certain that TA knows the incident IN, it is possible that the approver AP sends this information to TA using the message in which a reference to the claim is written. In this case some kind of covert or better: "private" channel is used between AP and TA. When TA combines the contents of IN with the knowledge of the age of SU, TR finds out that the incident concerned illegally driving a car because SU was too young.

3. The cashier CA seems to know only the amount to be paid and the submitter's bank and it seems that no privacy problem can occur, however, suppose the amount is very high, so that CA may want to know SU's name. By conspiring with the approver AP and the travel agent TA, using a private channel, CA can get SU's name indeed.

The three examples show that the analysis has to reveal what each employee knows about the individual, here the submitter SU. It also reveals that we need to know what each employee might know about the individual, by cooperating/conspiring with other employees, using private channels. So lists have to be made for each employee in an action what s/he knows according to the WF and what s/he may know additionally when another agent sends him/her a message with more information than the information prescribed in the WF.

4.1 About the verbs being used

We have to say something about the verbs describing the actions. Evidently, `send(ag=S, go=M, rec=R, ...)` means: S sends a message M to R. M may a tuple, as in `(trip TR, incident IN, amount AM1)` or an object identifier, as in: `go=CL`, or an unspecified thing as in `go=answer`. There are also verbs denoting some specific action as: "verify", "treat", or "pay". For these actions it is necessary that the contents of their parameter "goal" is known on the basis of "need to know". The "create" and "fill in" actions are also special as they bring new objects into being. For these actions detailed knowledge about their parameter "goal" is not needed. The computer can carry out these actions without the agent knowing these details.

COLOR-X is a system which has been designed with Linguistics in mind, in fact it is based on the linguistic theory Functional Grammar [Dik89, Bur96], while the proper working of the COLOR-X system requires a connection with a Lexicon, such as WordNet [Fel98], in which the meaning of words and concepts is stored. In our case we would like the Lexicon making the distinction between these different kinds of verbs: "send" is a communicative verb, while "treat" and "verify" are verbs connected with the notion of performing. To "create" and "fill in" are verbs connected to "making something new".

4.2 About the identifiers being used

In the WF many identifiers are being used and they are of different kinds. They denote objects, such as SU, the submitter, or CL a new object created to handle the claim. They also may denote attributes, usually with information about the submitter, in which case privacy may be a concern, or about the claim. Some identifiers are made within the WF, such as AN, denoting an answer used within a message, or TA being a shorthand for `TR.travel_agent`. One could think that only the identifiers belonging to information about the submitter are important for privacy. This is not the case. Also identifiers whose values are created in the WF can reveal important information about the submitter. For example, the amount of money AM2, which is to be paid, says something about the trip and accident of the submitter.

4.3 About the representation of the WorkFlow in Prolog

The representation of the WF is adapted to the above analysis; facts and functors are used for: nodes, processes, actions, roles, identifiers, constraints, comparisons, expressions, edges, types and attributes. As WFs can be run through

in different ways, these have to be administered in the form of flows. For our example there are five flows possible, given in figure 4. In general there may be an infinite number of flows, so in order for the analysis program not to loop, it is also necessary to find cycles in the WF.

The numbers in this figure are the index numbers of the nodes in the claim CEM, the arrows are transitions. If there are multiple possibilities for following a transition then this is indicated above the arrow. If there is only one possibility then this is notated as a single $\rightarrow$. Transitions that occur when a `MUST` condition is violated, are notated as $\rightsquigarrow$.

$$
\begin{array}{ll}
\#1 & 0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightsquigarrow 4 \rightarrow \text{-}1 \\
\#2 & 0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \xrightarrow{\texttt{AN="not OK"}} 4 \rightarrow \text{-}1 \\
\#3 & 0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \xrightarrow{\texttt{AN="OK" and AM1<100}} 5 \rightarrow 10 \rightarrow \text{-}1 \\
\#4 & 0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \xrightarrow{\texttt{AN="OK" and AM1>=100}} 6 \rightarrow 7 \xrightarrow{\texttt{ME4="not OK"}} 4 \rightarrow \text{-}1 \\
\#5 & 0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \xrightarrow{\texttt{AN="OK" and AM1>=100}} 6 \rightarrow 7 \xrightarrow{\texttt{ME4="OK"}} 8 \rightarrow 9 \rightarrow \text{-}1
\end{array}
$$

Figure 4: Flows of the claim CEM.

The different identifiers in the WF are represented by denoting the roles in which they occur in the event actions, such as ag (agent), rec (recipient) or go (goal).

In table 1 we see a list of the identifiers for the claim example. Let us analyze what the meaning is of this table. Take as example the information about the cashier CA in node 10:

9	CA	pay	goal AM2	SU, SU.name, TR
			rec SU.bank	(from AP, TA)

The second column indicates the identifiers which have to be known by the agent in order to do its work. Evidently, in order to be able to carry out the "pay" action, CA must have access to the contents of `AM2` and `SU.bank`. The approver AP in node 8, however, who sends the triple `(AM2, SU.bank)` object to CA does not need to know the contents of this triple, it is just sufficient to know its object identifier. He can send this identifier to CA as if it is put in a closed envelope. A special case is formed by identifiers like `AM2`, being sent by EX, as part of the message ME4. From the WF we cannot see what is the contents of this identifier. In fact it is the outcome of the "treat" action. Or it comes from the assignment `AM2=AM1`. The question is whether this identifier is also interesting from the standpoint of SU's privacy. It seems that it is harmless when it is used by the cashier CA. This is only seemingly the case: from the height of the amount the kind of claim and incident could be deduced.

The third column reveals that CA could know also `SU.name`, when namely the approver AP would help him. Indeed AP could know `SU.name` and send it unnoticed, using the message with the above triple, to CA. On its turn, AP needs TA to know `SU.name`. This is called a private channel, the existence of such channels is important as the second and third examples have shown, presented in the beginning of this section. For communicative actions, indicated by such verbs as: send, create and fill_in, only tokens are used to indicate identifiers of pieces of information, such as

node & agent	verb	role	identifiers	identifiers accessed using a private channel
1 SU	send	goal	GO1, TR, IN, AM1	
		rec	AP	
2 AP	create	goal	CL	
		obj	OBJ1, SU, TR, IN, AM1	
	send	goal	CL	
		rec	TA, TR.travel_agent	
3 TA	verify	goal	GO2, SU.name, TR	TR, IN, AM1
	send	goal	AN	(from AP)
		rec	AP	
4 AP	send	goal	ME1	SU.name
		rec	SU	(from TA)
5 AP	send	goal	GO3, AM1, SU.bank	SU.name
		rec	CA	(from TA)
	send	goal	ME2	
		rec	SU	
6 AP	send	goal	CL	SU.name
		rec	EX	(from TA)
	send	goal	ME3	
		rec	SU	
7 EX	treat	goal	GO4, TR, IN	SU.name, AM1
	send	goal	ME4	(from AP, TA)
		rec	AP	
8 AP	fill_in	goal	AM2	
		rec	CL	
	send	goal	GO5, AM2, SU.bank	
		rec	CA	
9 CA	pay	goal	AM2	SU, SU.name, TR
		rec	SU.bank	(from AP, TA)

Table 1: Access of agents to the identifiers in the claim CEM.

an object identifier or a pointer, and there is no threat for violating privacy rules. Only when also a right to read or to write is needed, this threat exists, as the contents of objects and messages (also objects in Mokum) is at stake. This is the case with actions such as "treat, verify and pay".

5 The WorkFlow analyzed

The Workflow can be analyzed for several reasons. They may have to do with rather general properties such as: every agent who needs certain information is provided with that information. This is a property concerning the quality of the WF. Our analysis program provides this type of analysis. For our interest in S&P we may see different aspects to be analyzed:

- Are the employees of a company provided with the information needed for their work (need-to-know) and don't have more information at their disposal, which may be contrary to privacy rules/laws.
- The Alter-ego in company A, when deciding to answer a query Q coming from an employee E of company B, can use the knowledge coming out of the analysis in this way: Is the knowledge of E together with the answer to Q enough to entail information about the individual that the individual does not want E to know.

The first aspect is important when analyzing whether the insurance company is making fair decisions, that is decisions which don't take the specific individual into account (such as the information that the submitter SU is a nephew of the approver AP) WF is analyzed whether a decision maker in WF is not using personal information, such as name or address, which he does not need. For our WF this analysis reveals that TA could make use of some personal information, namely SU's name. It can do that not directly, as its only information is the claim object CL, to which it does not have reading access, but because it has reading access to GO2 and because according to the constraint in node 3: `id GO2 =3D (SU.name, TR)`, `SU.name` is part of GO2, indeed TA can see SU's name. A small Prolog program for this analysis and its results are given in the Appendix.

The second aspect is demonstrated by the existence of private channels. We have seen in the second and third example how a private channel can be used to jeopardize privacy rules involving conspiring employees. It may be necessary that the management of the insurance company is notified that this type of possible conspiracy exists. In fact it may be the task of an auditor to signal these possible privacy breaches.

We have seen examples where the analysis of the WF leads to interesting results. The analysis is based on table 1 which gives for each agent in an action defined in a node two lists:

1. the identifiers it needs to know in order to do the work specified,

2. the identifiers it might know by conspiring with other employees (using messages for which they are not meant, so creating private channels).

From this table the Alter-ego for our individual can easily determine whether a certain query can be answered or not. Take the query in the first example in the beginning of this section asked by the travel agent TA about the individual's marital status. The table reveals that TA (needs to) know(s) the trip TR, so the individual may want his Alter-ego to refuse the correct answer. In the second example, TA is interested in the individual's age, and again, if the individual does not trust the TA he can refuse to answer the query.

5.1 Real covert channels

The use of private channels such as we introduced them in the preceding sections, can in principle be detected by the WorkFlow engine, i.e. the underlying machine which governs the carrying out of the WF. The messages sent by all agents involved can be inspected, so that it can be seen that agents are sending each other more pieces of information than is asked for. In the study we describe here another form of using real covert channels has been dealt with also. In this kind of covert channel an agent puts information in some piece of information s/he has produced him/herself, such as changing the amount AM2 from 2000 to 2000.22, where "22" is some predefined code. Or when the information is a character string using lower and uppercase letters in some predefined way. In another report [Tee99] this kind of covert channels is fully analyzed.

6 Conclusions

We have shown how tools can be built by means of which WF diagrams can be analyzed on privacy aspects. These tools can also be used to analyze other aspects such as quality of the WF specification. Future work has to be carried out around detection of other forms of covert channels. Also the integration of these tools in the Security part of ERP systems is an interesting area. In [RJG98, RJOS00] we have studied the automatic connection of Security rules to WF specifications, and a comparison of Security architectures in a structure-based approach and the capability-driven approach of an ERP system. In particular the communication aspects in Cyberspace and the security involved is being discussed in these papers.

References

[AH96] V. Atluri and W-K. Huang. An extended petri net model for supporting workflows in a multilevel secure environment. In *Proceedings of the 10th IFIP*

WG 11.3 Working conference on Database Security, pages 199–216, July 1996. http://cimic.rutgers.edu/~atluri/ifip96.ps.

[Bur96] J.F.M. Burg. *Linguistic Instruments in Requirements Engineering.* PhD thesis, Department of Mathematics and Computer Science, Vrije Universiteit Amsterdam, 1996.

[Dik89] S.C. Dik. *The Structure of the Clause*, volume 1 of *The Theory of Functional Grammar.* Floris Publications, Dordrecht, 1989.

[Fel98] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database.* MIT Press, Cambridge, MA, 1998.

[GHS95] D. Georgakopoulos, M. Hornick, and A. Sheth. An overview of workflow management: from process modelling to workflow automation infrastructure. *Distributed and Parallel Databases*, 3(2):119–154, 1995.

[GRBO97] E. Gudes, R.P. van de Riet, J.F.M. Burg, and M.S. Olivier. Alter-egos and roles – supporting workflow security in cyberspace. In *Proceedings of the IFIP WG 11.3 Database Security Conference (DBSec'97)*, Lake Tahoe, USA, 1997.

[RJG98] R.P. van de Riet, J. Janssen, and P. de Gruijter. Security moving from database systems to erp systems. In R. Wagner, editor, *Database and Expert Systems Applications*, pages 273–280. IEEE Computer Society, 1998.

[RJOS00] R.P. van de Riet, W. Janssen, M. Olivier, and Radu Serban. A comparison of two architectures for implementing security and privacy in cyberspace. Technical report, Department of Mathematics and Computer Science, Vrije Universiteit Amsterdam, February 2000.

[Tee99] Wouter Teepe. Privacy-gerichte workflowanalyse. Master's thesis, Rijksuniversiteit Groningen, 1999.

[vdRB96a] R.P. van de Riet and J.F.M. Burg. Linguistic tools for modelling alter egos in cyberspace: Who is responsible? *Journal of Universal Computer Science*, 2(9):623–636, 1996. http://www.iicm.edu/jucs_2_9/linguistic_tools_for_modelling/.

[vdRB96b] R.P. van de Riet and J.F.M. Burg. Modelling alter egos in cyberspace: Who is responsible? In *Proceedings of the World Conference of the Web Society (WebNet'96).* AACE, 1996. http://curry.edschool.virginia.edu/aace/conf/webnet/html/210.htm.

A Appendix: Some Prolog and results of the analysis

A small Prolog program concerning the computation of what the agents can see as derived from the WorkFlow diagram is presented here just to give the flavor of how these kinds of analysis looks like.

```
demo1(Handle,Method,Datas,AlertType,RedAlerts) :-
  findall(alert(FE,Index,Created,RedAlert),
    (create(Handle,Created,Index,Agent),
     flowedge(Handle,Method,Index,FE,_,_,_),
     collect_knowledge(Handle,Method,FE,Agent,Accesses),
     member(ThisData,Datas),
     member(RedAlert,Accesses),
     RedAlert = access(_,_,identifier(_,Data),_,Type),
     subset(AlertType,Type),
     append(ThisData,_,Data)
    ),RedAlerts).
```

Figure 5: Example Prolog program

The results of this program are given now, from which you can see that TA knows `SU.name`.

```
?- demo1(cl,2,[['SU']],[id,cont],RedAlerts).

RedAlerts = [
  alert(fe4, 3, ['AN'], access([
      add(fe4, identifier([], ['SU', name]), apparentlyknow(['TA'])),
      add(fe4,
        tuple([identifier([], ['SU', name]), identifier([], ['TR'])]), tuple),
      add(fe4, identifier([], ['GO2']), constraint(3, 1)),
      add(fe4, identifier([], ['GO2']), action(3, 1, 1))
    ], ['TA'], identifier([], ['SU', name]), card, [id, cont]))] ;

No
?-
```

Figure 6: Conflicting access for example 1.