A Formal Security Model for Collaboration in Multi-agency Networks

Salem Aljareh [1], Nick Rossiter [2] and Michael Heather [3]

[1] Computing Science, Newcastle University, Newcastle upon Tyne, UK NE1 7RU

[2] Informatics, University of Northumbria, Newcastle upon Tyne , UK, NE1 8ST

[3] Law, University of Northumbria, Newcastle upon Tyne , UK, NE1 8ST

Abstract. Security problems in collaborative work between multiple agencies are less well understood than those in the business and defence worlds. We develop a perspective for policies and models that is task-based on a need-to-know basis. These policies are represented by two protocols, the first CTCP (Collaboration Task-based Creation Protocol) dealing with negotiation, decision and agreement between the parties involved and the second CTRP (Collaboration Task-based Run-time Protocol) responsible for the operation of the policy. The two protocols and the relationship between them are defined in Petri-Nets. The overall model is formally defined using a categorical pullback construction. Each of the protocols, represented as Petri-Nets for state-transition purposes, is a category-valued functor in the pullback.

1 Introduction

Information is naturally sharable among groups such as team, committee, organization, country and federation in a manner based on trust. However to achieve an accepted level of trust is quite a complicated issue because as the collaboration grows wider, more participants are involved with divergent policies. Although designing secure models for collaboration environments has been a target of a number of academic and commercial research bodies, numerous organizations still keep their systems (especially the trusted systems) unconnected with outsiders.

Basically security systems are built out of the available mechanisms to meet a security policy based on a selected security model [9]. A review has been made [2] of the appropriateness of standard security models for collaborative multi-agency systems. Most are either targeted at a specific security requirement or are too static to represent a dynamic situation. All deal with a single policy, whereas by definition the multi-agency and collaboration environment involves more than one policy.

A motivating example of an application that involves multi-agency services is the medical information services. The only model designed to meet the security requirements for the medical records in the UK was the BMA (British Medical Association) Security Policy Model [3]. This model was recently examined [1] against the multi-agency security requirements and it was found that the issue of sharing clinical information including collaboration activities with other agencies such as police, social services or the education authority was not clearly considered. For instance the *need-to-know* problem was not addressed in the BMA model, as the

Aljareh S., Rossiter N. and Heather M. (2004).
A Formal Security Model for Collaboration in Multi-agency Networks.
In *Proceedings of the 1st International Workshop on Computer Supported Activity Coordination*, pages 253-260
DOI: 10.5220/0002681402530260

BMA does not accept that *need-to-know* is an acceptable basis for access control decisions. However there might be a case where *need-to-know* cannot be avoided. For example a service provider such as an insurance company offers its services conditioned by some information about the patient who applies for such services. An example is given in [1].

In this paper, we propose a security model that we argue will alleviate the security difficulties that may arise in attempts to build a collaboration network. The model is constructed from a task-based perspective, as this approach seems to offer the best way forward, as discussed later. An example of a prototype for informal collaboration, handled using the model, is given elsewhere [2]. The general principles of the model are discussed and a diagrammatic notation is devised. Two task-based collaboration protocols, expressed in this paper in the form of Petri-Nets, represent the permitted states and transitions. The choice of Petri-Nets as the notation is discussed. Finally the overall model is constructed formally as categorical pullbacks to illustrate its foundation on established logical principles.

2 A Task-based Perspective for Collaboration Networks

A collaboration business, by definition, is based on the needs of the collaborators from each other. Each side needs information or a service from the other participants. The obvious question that someone will immediately ask before he/she releases any confidential information or responds to an enquiry is: what for? For what purpose is the information required? Usually the expected answer will be the naming of a task for which the information required is essential, sometimes with a further explanation of the benefit of this task for the two sides (collaboration proposal). The information owner may like to restrict the use of this information by some conditions (security policy). If they reach initial agreement a detailed negotiation will then take place until they reach a considered level of trust, which leads to a collaboration agreement to perform the task. One reasonable condition might be to limit the use of the information by other tasks. For instance it could be specified that the information should not be used outside the task for any purpose.

We decided to construct our model as task-oriented for the following reasons:

1. Fundamentally any collaboration scheme is based on specific tasks: there is no collaboration without a task.
2. The task-based approach is promising to address the need-to-know problem, satisfying a user requirement in any multi-agency services environment.
3. The collaboration task is the common object between the collaborators.
4. Shared information ownership can be granted to the collaboration task.
5. The task is scalable, flexible and dynamic.
6. Explicit responsibility is recognized in the task-based approach.

Overall the basis for any collaboration is an aim to share resources in order to achieve common benefits by performing shared operations. Other task-based approaches to security are discussed later.

3 General Principles for our Model

Collaboration: In our model we consider any deal/trade between individuals or groups, which aims to benefit the sides involved as a kind of collaboration. The following are some forms of collaboration:

Trading between customers and service providers.

Joint operation projects

Research group collaboration.

The clinician and the patient trade/relationship: the clinician's job exists because of the patient, and the patient needs the clinician for treatment. So both need each other and benefit each other. The clinician may need to know some information from the patient as part of the course of treatment. The relationship is in general based on trust. In this example there are two sides trading benefits through the task called treatment

Ownership: In this model an item of information is owned initially by its natural owner that is the person to whom the information relates. For instance information about the baby is owned by the baby although this information is controlled by guardian/parents. In computer security terms this is called *grant access* or *delegation.* Once this information is required to be shared among collaboration parties, an access will be granted to what we call the *collaboration-task*, controlled by the *task-policy*. The information owner and/or the access controller will be part of the negotiation that results in the task policy.

Authorization: A participant in a collaboration network, called task-participant, will be authorised to gain access to a collaboration-task. This authority will be limited by what we call task-policy.

3.1 Collaborative Task Characteristics

The characteristics of collaborative tasks are considered to be:

1. Flexible: can be a single activity or group of activities sharing same policy, each of which can be selected as the need arises.
2. Dynamic: can be updated even while it is running (supporting post-hoc justification). For instance a nurse can be replaced by another one if he/she is not, for any reason, able to complete his/her duty in a surgical operation. However any change in the task elements should be fully and carefully documented.
3. Secure: should be fully protected using all the available mechanisms.
4. Scalable: can be upgraded, for instance to fill some gaps in the original task. A new collaboration task can be built starting from default tasks.
5. Accountable: all collaboration protocol states and all task run-time events of the collaboration must be well documented.

3.2 Diagrammatic Representation of Model

The architecture in Figure 1 illustrates the general components of our model. The main component is the collaborators (two or more), each of which will need to define three elements: requirements (what does he/she/it/they aim to gain from the other side), policy (rules that need to be obeyed) and material (e.g. information to release or services to offer). The second component is a pair of task-based collaboration protocols -- the Collaboration Task Creation Protocol (CTCP) and the Collaboration Task Runtime Protocol (CTRP) -- both detailed later in the following sections.

CTCP includes a negotiation between all collaborators where the proposed task will be discussed including all collaborators' policies and requirements. This process (negotiation) continues until a decision is taken either by rejecting the proposal or by accepting it. The acceptance of a proposal will lead to a formal agreement/contract, which will produce the proposed collaboration task in its final stage including all of the policies and requirements. Negotiation can of course be a very complex task [5]. The work described here could be extended later to include such aspects as conflict resolution. CTRP will start after a successful compilation of CTCP and as scheduled in the *task_policy* (not necessarily immediately after the end of CTCP).

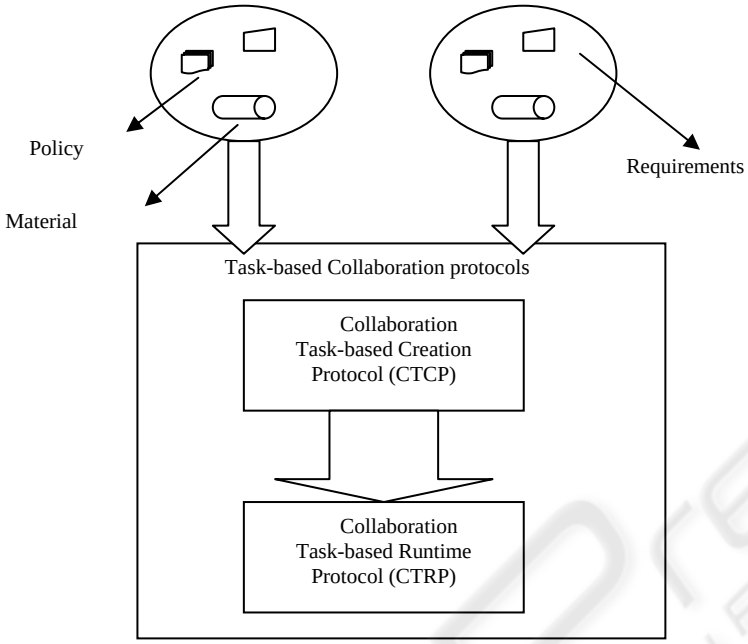

Fig. 1. General Architecture for Secure Collaboration Environment

(Requirements, offer, policy) * No of Collaborators

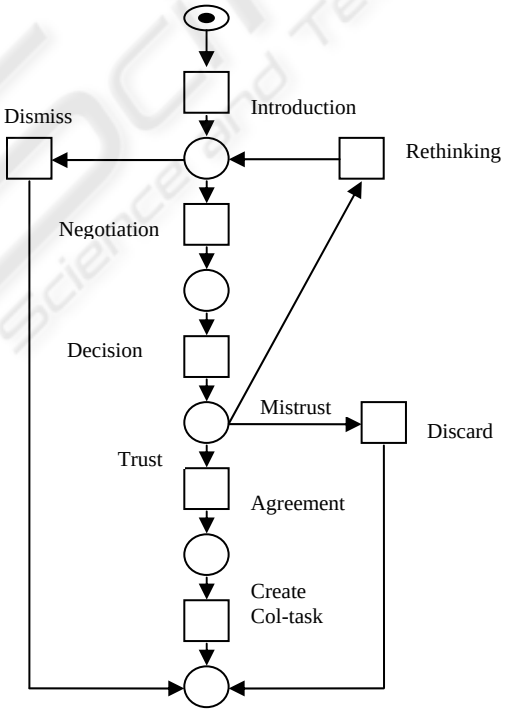

Fig. 2. Petri-Nets Graph: Task Creation Protocol

4 Representation of Protocols in Petri-Net Notation

We use the Petri-Nets model to represent our collaboration protocols to provide a formal basis and a more applicable medium for computer scientists. Flow charts lack a formal basis and can be ambiguous in representing states and transitions. Data flow diagrams emphasise flows of data, not states, which are considered critical in security systems.

Net theory was originally introduced in a PhD thesis of C. A. Petri. Later Reisig [15] introduced it to the software engineering area. More recent advances in this formalism are described in [16]. The usefulness of Petri-Nets in providing a theoretical basis for handling object life cycles has been demonstrated by [22]. In collaboration networks, similar to the multi-agency services investigated here, Furuta and Stotts [8] presented an evolution of the Trellis model by providing a formal Petri net basis for prototyping the control of such a network.

In the security area an industrial use of Coloured Petri-Nets was developed by [14] making it possible to perform simulations. The nets were debugged by constructing reachability graphs. Joshi and Ghafoor [11] specified a multi-level security model for multimedia using a time-augmented coloured Petri-Net model. For cryptographic protocols [6] use Petri-Nets to illustrate how their semantics can be used to prove security properties. Ryan [18] notes that causality, critical in the analysis of security protocols, is closely related to information flow and that causal structures are rather more explicit in Petri-Nets than in many other areas.

In general Petri-Nets have been widely used for the modelling and analysis of systems that are characterized as being concurrent, asynchronous, distributed, parallel and non-deterministic [10]. All these features apply in the collaborative, multi-agency systems studied here. Activities in the systems: a) overlap in timing; b) are run independently rather than according to some common time signal; c) are run over many different servers; d) involve the splitting of tasks into subtasks which run in parallel until some common join point is reached; and e) may not give the same result in negotiation each time the protocols are run.

The Petri-Net in Figure 2 represents the CTCP protocol. The initial state represents for each collaborator their requirements, policies and offers. For instance, in the patient-doctor collaboration, the patient's requirements are treatments, the patient's policy is to keep personal information secret, the doctor's requirements may include information about the patient and the doctor's offer is a treatment course. Following discussion of this initial state the task, at first an offer from one side or a requirement from another, is accepted as an offer for further negotiation or rejected without any further details. Policy considerations are normally omitted during the introduction transition.

If the proposed task is found to be reasonable then all collaborators will enter into a detailed negotiation in which all aspects including requirements, services and polices will be clarified for all collaborators. After that one of three decisions will be taken: the first option could be one of the collaborators needs more time to think about the task/offer; the second option could be that the expected level of trust could not be ensured so the task is simply dismissed; the third option is that all collaborators trust each others so that an agreement between all collaborators will take place. This agreement at the end will be formulated in what we call the collaboration task. This task will be limited in scope by the task policy, which is a composition of all collaborators' policies, meeting all sides' requirements.

The Collaboration Task Runtime protocol (CTRP) starts after the task has been completely created by the CTCP protocol and when its schedule time, according to the task-policy, is due. Before starting the process of the task some tasks need some preparations. Then the task process starts following the policy that has been approved in the CTCP stage. Each state of this process is monitored, assessed (verified against the task-policy) and then documented. The task assessment may result in one of the following:

1. The task is proceeding satisfactorily, following the policy and the plan and has not finished yet, so the task should persist.

2. The task needs an update to meet its requirements. Depending on how the updates affect the process: the task may restart or continue from the last process state.
3. The task reaches its scheduled end; hence the task terminates normally.
4. There might be a case where the task abnormally terminates, for instance the task-policy has been violated, or the task exceeds the scheduled time without valid reasons. The abnormal termination could lead either to the end of the task and then the collaboration or to a new session of the CTCP. An exception is raised when the policy has been violated

In our model exceptions are divided into three types:

1. Exceptions with which the task can still continue to its normal end. Exceptions of this type are handled within the CTRP protocol by the task update component.
2. Exceptions with which the task must be terminated and another task is required to complete the planned function. Such cases are handled partially in the CTRP protocol. The task in such cases is aborted and the process log (task history) used by the CTCP protocol to create another task to redo the function that could not be done by the terminated task in view of the exceptions that have arisen.
3. Exceptions with which the task must be terminated and there is no need for any further actions. There are cases where the task immediately terminated and no further actions are possible. Exceptions from this type are handled within the CTRP protocol through the ABORT component.

5 Formalisation with Categorical Pullbacks

The relationship between the protocols CTCP and CTRP can be represented rigorously by the categorical pullback shown in Figure 3. Pullbacks are examples of cartesian closed categories [12].

This figure shows the relationship between four categories (denoted in bold font). **C** is the complete environment and **A** is a particular system to which a user may require access. **C/B** is a slice category or subcategory of **C** and **C** $\mathbf{X_B}$ **A** is a limit, representing the relationship between **C** and **A** in the context of **B**. The limit can be viewed as a subcategory of the product **C X A** over **B**. Three functors map between **C** $\mathbf{X_B}$ **A** and **C/B.** $\exists$ is the existential quantifier selecting some **C/B** for a particular **C** $\mathbf{X_B}$ **A,** $\forall$ is the universal quantifier selecting **C/B** that satisfy all the rules determined by Δ as the diagonal functor selecting a limit **C** $\mathbf{X_B}$ **A** for a particular subcategory **C/B**. Δ is right adjoint to $\exists$ and left adjoint to $\forall$, written $\exists$ -| Δ -| $\forall$. Two natural transformations are shown. η_c is the unit of adjunction comparing objects C with objects C X_B A and ε_{cxa} is the counit of adjunction comparing objects C X_B A with objects A. η_c is an inverse projection (π^*) and $\varepsilon_{c\,x\,a}$ is a projection (π).

In terms of our CTCP/CTRP model given above: the diagonal functor Δ corresponds to the protocol CTCP whereby a limit **C** $\mathbf{X_B}$ **A** is selected for a particular purpose **C/B** through negotiation. CTCP selects a relationship between C and A for a particular purpose such that the diagram in Figure 3 commutes, that is ι_c o $\pi_{c\,x\,a} = \iota_a$ o $\varepsilon_{c\,x\,a}$. As a Petri-Net CTCP can be represented as a monoidal category [4]. CTCP is therefore a category-valued functor. **C** $\mathbf{X_B}$ **A** corresponds to the policy rules derived through the negotiation in CTCP.

The existential functor $\exists$ is a type constraint: there must exist for all policy rules in **C** $\mathbf{X_B}$ **A** an entry in the system **C/B**. The universal quantifier functor $\forall$ corresponds to the protocol CTRP: all the rules held in the negotiated policy (the limit **C** $\mathbf{X_B}$ **A**) are applied for a particular purpose (**C/B**). Like CTCP, CTRP is a category-valued functor with its Petri-Net defined as a monoidal category. Overall CTCP is right-adjoint to $\exists$ and left-adjoint to CTRP. CTRP is right-adjoint to CTCP.

Exceptions are much less likely to occur in the strongly typed categorical model than in a set model. If they did occur they would be handled at the natural transformation level. The unit of adjunction η_c is given as $1_c \rightarrow$ CTRP o CTCP(c) and the counit of adjunction as CTCP o CTRP(c x a) $\rightarrow 1_{c\,x\,a}$. The former measures the change in c as the functors CTCP and CTRP are applied in turn. The latter measures the change in (c x a) as the functors CTRP and CTCP

are applied in turn. The unit and counit both give a measure of consistency as the application is run with the possibility of exceptions being raised if divergence is noted.

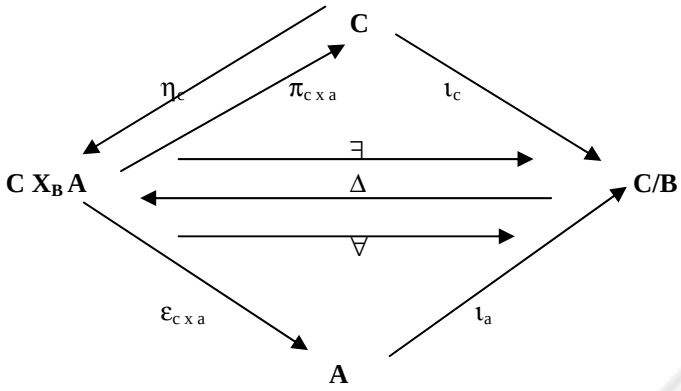

Fig. 3. Categorical Pullback of System (**A**) over Environment (**C**) in the context of Purpose/View (**C**/**B**)

6 Discussion and Conclusions

We consider two aspects of our work. Firstly the extent to which task-based approaches have been used before in security systems; secondly the prospects for formal approaches in the security area.

The idea of task-based has been introduced before in a number of models [7,19,20]. All were at the basic level of this approach. The focus by the last two [19,20]] was on whether a task-based security model could be an alternative authorisation and access control model to the subject-object traditional authorisation models. The first paper [7] tried to address the privacy problem using the task-based approach. Mahling *et al* [13] tried to build a task-based collaboration model. This work though starts from a relatively late stage in the negotiation where the plan, agreement and tasks are relatively clear. In addition they do not consider the case of the multi-agency environments where the policies of the collaborators are different.

We have intended in our model to use all of the power of this idea (task-based approach) to address the security problem of the collaboration networks and the multi-agency services environment. We argue that the real challenge for the task-based approach is the multi-agency services environment, where responsibilities are distributed and the ownership is dynamic. None of the existing approaches address multi-agency aspects in detail.

Formalising security models is an important matter as this a way in which guarantees can be secured about the reliability of a model. Security rules for multi-agency systems need to be formulated at the policy level. At this level, category theory seems to be appropriate as it provides not only appropriate abstractions for this level but also, in a multi-level architecture, mappings to lower levels such as mechanisms. For interoperability a multi-level approach constructed in category theory has already proved very promising [17]. The use of Petri-Nets, for detailed state transitions, within a categorical framework, for control of types and levels, looks to be a way forward for formalising security in information systems. More advanced techniques such as Timed Petri-Nets and Stochastic Petri-Nets should be useful in gaining greater expressibility. Validation techniques in Petri-Nets could also be used for verifying the model. The benefits of using Petri-Nets will be highest where collaboration occurs between multiple agencies. This is a natural area for applying Petri Nets with its concurrency, asynchronicity, distribution, parallelism and non-determinism.

To conclude this paper has introduced a task-based model to facilitate collaboration in trusted multi-agency networks. Our model is based on the fundamental aspect of the collaboration environment, which is the task-based perspective. Two task-based collaboration protocols (CTCP and CTRP), expressed as Petri-Nets, are used to represent the permitted states and

transitions. The extent to which task-based approaches have been used before in security systems has also been discussed.

The two protocols and the relationship between them are defined in Petri-Nets. The overall model is formally defined using a categorical pullback construction. Each of the protocols, represented as Petri-Nets for state-transition purposes, is a category-valued functor in the pullback. The use of Petri-Nets within a categorical framework looks to be a promising way forward for security problems.

References

1. Aljareh, S., & Rossiter N., 2001, Toward security in multi-agency clinical information services, *Proceedings Workshop on Dependability in Healthcare Informatics*, Edinburgh, 22nd-23rd March 2001, 33-41.
2. Aljareh, S., & Rossiter, N., 2002, A Task-based Security Model to facilitate Collaboration in Trusted Multi-agency Networks, *ACM Symposium on Applied Computing (SAC) 2002*, Madrid, 744-749.
3. Anderson, R., 1996, A Security Policy Model for clinical Information Systems, *Proc. IEEE Symposium on Research in Security and Privacy*, 30–43.
4. Asperti, A., Ferrari, G. L., & Gorrieri, R., 1990, Implicative formulae in the `Proofs as Computations' analogy, *Proc 17th ACM SIGPLAN-SIGACT Symp Principles Programming Languages*, 59-71.
5. Chu-Carroll, J., and Carberry, S., 2000, Conflict Resolution in Collaborative Planning Dialogues, *International Journal of Human-Computer Studies*, 53(6) 969-1015.
6. Crazzolara, F., & G. Winskel, G., 2001, Petri-Nets in cryptographic protocols, *Proc. 6th Inl Workshop Formal methods Parallel Programming: Theory and Practice*.
7. Fischer-Hübner, S., & Ott, A., 1998, From a Formal Privacy Model to its Implementation, *Proc. 21st National Information Systems Security Conference*, Arlington, VA.
8. Furuta, R, & Stotts, P D, 1994, Interpreted collaboration protocols and their use in GroupWise prototyping, *Proc 1994 ACM Conf Computer supported cooperative work*, Chapel Hill, North Carolina, United States, 121 – 131.
9. Gollmann, D., 1999, *Computer Security*. ISBN: 0 471 97844 2, John Wiley and Sons.
10. Jensen, K., 1996, *Colored Petri-Nets - Basic concepts, analysis methods and practical use*, Springer, second edition **1**.
11. Joshi, J., & Ghafoor, A., 2000, A Petri-Net Based Multilevel Security Specification Model for Multimedia Documents, *ICME2000, IEEE International Conference on Multimedia and Expo*, MP10.12 533, Purdue University, USA.
12. Mac Lane, S, 1998, *Categories for the Working Mathematician*, 2nd ed, Springer-Verlag.
13. Mahling, D.E., Coury, B. G., & Croft, W. B., 1990, User Models in Cooperative Task-oriented environment. *Proc. 23rd Annual Hawaii IEEE International Conference on System Science*, 94-99.
14. Rasmussen, J. L., & Singh, M., 1996, Designing a Security System by Means of Coloured Petri-Nets. Proc. 17th International Conference in Application and Theory of Petri-Nets (ICATPN'96), Osaka, Japan, *Lecture Notes in Computer Science*, **1091** 400-419.
15. Reisig, W., 1985, *Petri-Nets: an Introduction*. Berlin; New York: Springer-Verlag.
16. Reisig, W., & Rozenberg G., 1998, Lectures on Petri-Nets: Advances in Petri-Nets. *Lecture Notes in Computer Science*, no. 1491.
17. Rossiter, N., Nelson, D. A., & Heather, M. A., 2003, Formalizing Types with Ultimate Closure for Middleware Tools in Information Systems Engineering, *5th International Conference on Enterprise Information Systems (ICEIS)*, Angers, France 366-373.
18. Ryan, P, 2003, Theoretical Challenges Raised by Information Security, *Workshop on Issues in Security and Petri-Nets (WISP), ICATPN*.
19. Steinke, G., 1997, A Task-based Approach to Implementing Computer Security, *Journal of Computer Information Systems*, 47-54.
20. Thomas, R. K., & Sandhu, R. S., 1994, Conceptual Foundation for a Model of Task-Based Authorization, *Proc. 7th IEEE Computer Security Foundations Workshop*, 66-79.
21. Thomas, R. K., & Sandhu, R. S., 1997, Task-based Authorization Controls (TBAC): A Family of Models for Active and Enterprise-oriented Authorization Management. *Proc. IFIP WG11.3 Workshop on Database Security*, Lake Tahoe, California.
22. Van der Aalst, W. M. P., & Basten, D., 2001, Identifying Commonalities and differences in Object Life Cycles using Behavioral Inheritance, Application and Theory of Petri-Nets 2001, *22nd International Conference ICATPN*, Newcastle, 32-52.