Modelling Collaboration in Trusted Multi-agency Environment

Salem Aljareh and Nick Rossiter
Computing Science, Newcastle University, UK, NE1 7RU
{s.s.aljareh, b.n.rossiter}@ncl.ac.uk

ABSTRACT
Security problems in collaboration work are less well understood than those in the business and defence worlds. Earlier work in the medical area in Britain has highlighted some of the principles involved but has neglected some important areas. Most of the security models developed to date are inadequate in the collaboration area. We develop a perspective for policies and models that is task-based on a need-to-know basis. These policies can be represented as Petri nets to identify the functions and states involved. We have also developed a general architecture for a secure collaborative environment. An example is given of the application of our techniques to a problem involving negotiation, decision and agreement in a collaboration environment.

Keywords
Collaboration networks, security models, multi-agency service.

1. INTRODUCTION

Information about an individual is absolutely secure, as long as nobody else has access to it, which is true only in the case where an individual is completely independent. As a consequence 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 and several works have been done (both theoretical and practical), still numerous organizations 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 [10]. The Bell-Lapadula Model (BLP) was the first formal model [4]. It is a state machine capturing the confidentiality aspect of the access control. It has been considered as a base or (at least a benchmark) for most security models and is used to design most of the available operating systems to support access control. Security policy in BLP is based on security levels and an access control matrix, which supports the traditional access rights (read, write, execute, append) but not the more recent access rights such as those required in a collaboration environment (viewing, coupling) [16]. However BLP deals only with confidentiality, not with the other computer security aspects such as integrity and does not address the management of access control as well as it has been proven that BLP could contain a convert channel[1]. To conclude the BLP model is suitable for an environment where polices are static.

[1] Convert channel is an information flow that is not controlled by security mechanisms.

Subsequent models attempted to fill the gaps left open in BLP. For instance the Harrison-Ruzzon-Ullman Model (HRU) [11] defines an authorisation system to address the problem of changing access rights, which was not addressed in BLP. In contrast to BLP, where access rights are usually assumed to be static, in the Chinese Wall Model [6] the access rights have to be re-examined at every state transition, to avoid conflict of interest. The Biba Model [5] deals with the integrity aspect. The Clark-Wilson Model [8] considers security requirements for commercial applications. It has two mechanisms to enforce integrity: the well-formed transaction and the separation of duties. The Information-Flow Model deals with the problem of the convert channel that has not been addressed by BLP and considers a system as a secure system if there is no illegal information flow. It should be noted that the above models are similar structurally.

Most of the security models that were designed subsequently were targeted at a specific security requirement. For instance multi-agency services and collaboration networks are based to some extent on these general models. However all these models are dealing 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 confidentiality requirements for the medical records in the UK was the BMA (British Medical Association) Security Policy Model [3]. This model was recently examined [2] 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 considered for policy reasons. For instance the *need-to-know* problem was not addressed in the BMA model, as the 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 instance 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 [2].

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. The general principles of the model are discussed and a diagrammatic representation is devised. Two task-based collaboration protocols, expressed in this paper in the form of Petri Nets, represent the permitted states and transitions. An example of informal collaboration is used to illustrate the application of the model.

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 have decided to build our model as a task-oriented model [1] 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

3.1 Collaboration

In our model we consider any deal/trade between individuals or groups which aims to benefit the sides involved is 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.

3.2 Ownership

The ownership is considered as a political issue and it is not only difficult to model but even difficult to define who own what. The following ownership aspects need to be understood well:

1. Right to own: fundamentally according to the general principles of freedom and human rights such as the General Declaration of Human rights 1948 an individual or group reserve the right to own their properties including personal information.
2. Ownership limitation: nevertheless an individual ownership is occasionally limited for the community/nation's benefit.
3. Regulation effect: granting or limiting the ownership is based on regulation by the law.
4. Ownership delegation: to meet the law and regulation requirements the ownership can be delegated but only under explicit principles and for a specific purpose.

An item of information, in this model, 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* and will be 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.

3.3 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*.

3.4 Responsibilities

All responsibilities should be explicitly defined in the task policy in the way each individual collaborator (*task-participant*) knows their responsibilities such as the required duties, the rules to follow (including ethical codes), the limitations (e.g. time, use of material and information) and the penalties.

4. COLLABORATION TASK CHARACTERISTICS

The following properties are required for a collaboration task:

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. (Accountability).
3. Secure: should be appropriately protected using all the available mechanisms.
4. Scalable: can be upgraded, for instance to fill some gaps in the original one. A new collaboration task can be built starting from default tasks (task template).
5. Accountable: all collaboration protocol states and all task run-time events of the collaboration must be well documented.

5. 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.

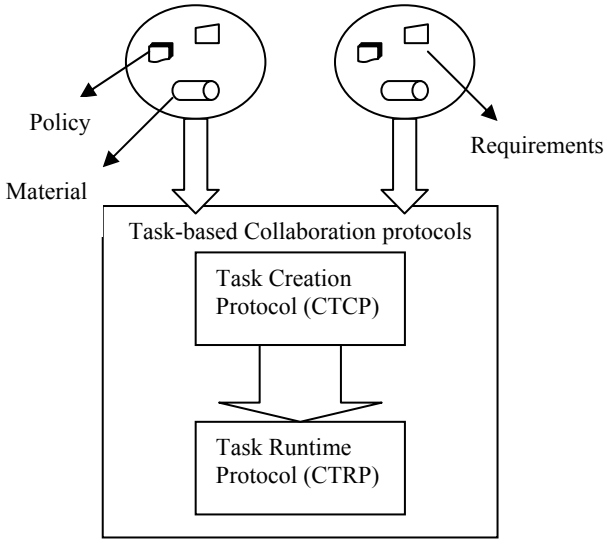

Figure 1: general architecture for secure collaboration environment

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 [7]. 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).

The main function of CTRP is to process the task that was previously created by the CTCP protocol and ensure that the *task_policy* is obeyed, the collaborators are aware of the circumstances and the right action is taken. CTRP is detailed in the following sections. In a special case of the abnormal termination of the task process the collaborators may need to go back to the CTCP protocol to create an alternative task. It should be noted that the *task_participants* (collaborators) are not necessarily to be the same subjects who were participating in the CTCP. However that should be included in the *task_policy*. The case of an emergency update for the participants list during the CTRP will be covered by the CTRP process documentation (the CTRP log)

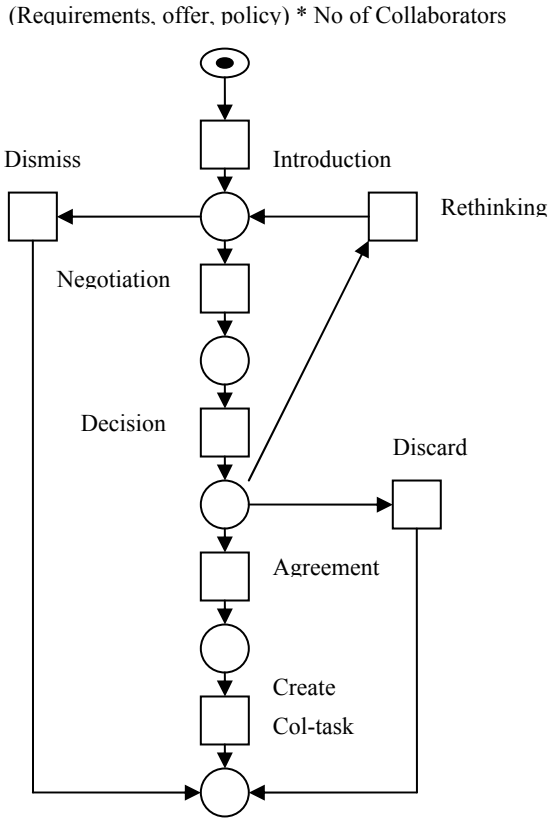

Figure 2: Petri Nets Graph representing the Collaboration Task Creation Protocol (CTCP)

6. FORMALISATION OF PROTOCOLS

We use the Petri Nets model to represent our collaboration protocols to provide a formal basis and to make it more readable for computer scientists. Flow charts lack a formal basis and can be ambiguous in representing states and transitions. Net theory was originally introduced in a PhD thesis of C. A. Petri and Reisig [15] introduced it to the software engineering area in 1985. More recent advances in this formalism are described in [14]. The usefulness of Petri Nets in providing a theoretical basis for handling object life cycles has been demonstrated by Van Der Aalst and Basten [20]. The Petri Net in Figure 2 represents the CTCP protocol.

The initial state represents the aim of each collaborator including 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. These aspects will be initially discussed as to whether the task (at first an offer from one side or a requirement from another) is accepted as an offer or rejected without any further details. The introduction transition will not include discussion about the policies. 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 that 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.

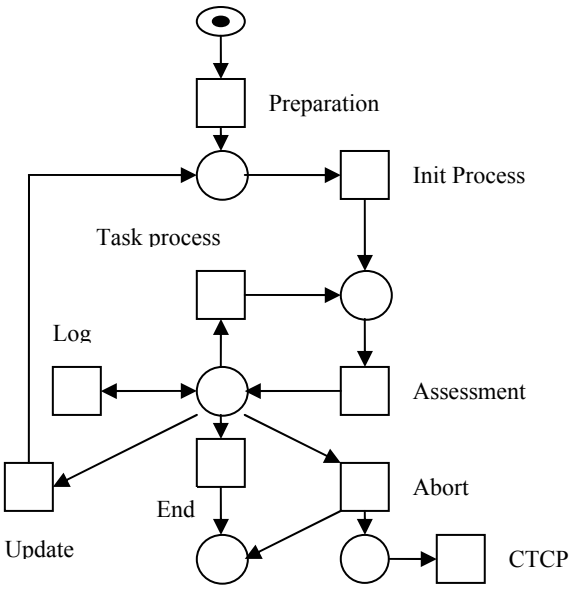

Figure 3: Petri Nets Graph representing the Collaboration Task Run-time Protocol (CTRP)

The Task Runtime protocol (CTRP), illustrated in Figure 3, 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 state of the process.
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 of the collaboration or to a new session of the CTCP.

7. EXAMPLE OF INFORMAL COLLABORATION

Let us consider a situation of a son asking his father for some cash:

7.1 The Collaboration Task Creation protocol (CTCP)

7.1.1 Introduction:

Son: father, I need 20 pounds [requirements].

Father: what for?

Son: to buy a book. [Purpose]

Father: well I do not have enough cash and I cannot drive to the ATM at the moment. [Initial discussion]

Son: Would you please lend me any of your cards (Debit or Credit) with the PIN, so I can go myself? [Proposed task]

7.1.2 Negotiation:

Father: well, you understand that you will not use this card for any other purpose, you will not withdraw more than 20 pounds, and you will not give away the card or its PIN to anybody else [Policy].

Son: Yes, I do understand that [accepting policy].

7.1.3 Decision:

Father based on his experience with his son will go for one of the following three options:

1. Take more time to think about the matter and to ask more questions. [Back to negotiation],
2. Cannot trust his son, so he cannot give him his card. [Dismiss the task], or
3. Trust his son and give him the card [commit the task].

7.1.4 Agreement:

- Father: I agree to give you my card along with the PIN but you should remember that:
- You return the card to me within 20 minutes of obtaining the money.
- You will not withdraw more than 20 pounds and you will not use the card for any other purpose.
- You will use the money to buy a book.

- You should not give the card nor disclose the PIN to anybody else.
- This agreement is based on trust between us.
- Son: Yes, I do understand all these conditions.

7.1.3 Rejection:

Father: I will go myself later to obtain for you the money that you want.

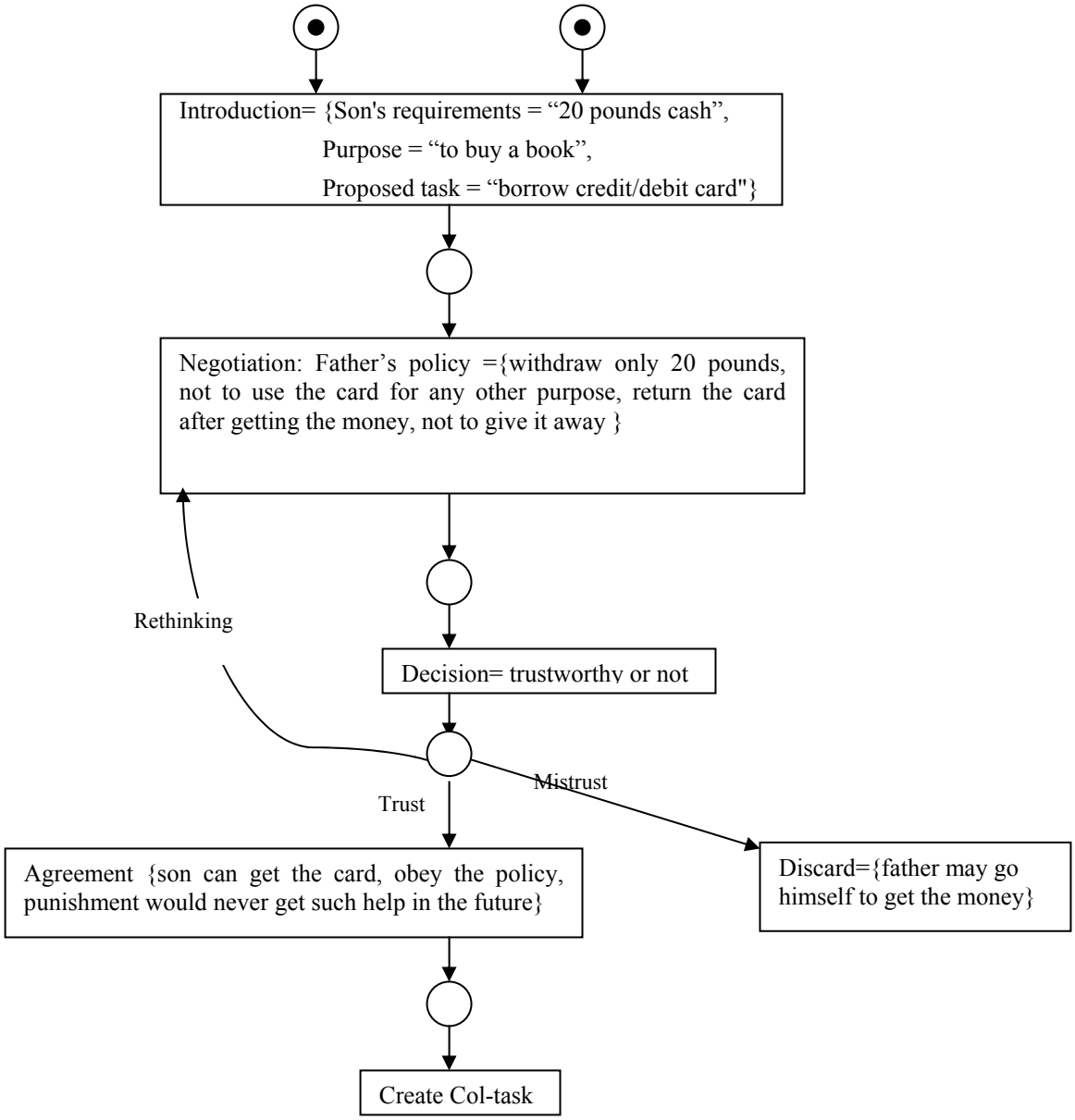

Figure 5: Petri Net representing the CTCP of the given example

7.2 The Collaboration Task Runtime Protocol (CTRP).

7.2.1 Preparation:

Farther: Explains to his son how to use the card and gives it to him.

7.2.2 Task process:

Son takes the card and starts using it.

7.2.3 Task assessment:

Father watches the time, maybe checks his account with another card if it takes more time than expected and takes decisions accordingly. Meanwhile he updates his relationship of trust with his son in the light of this experience.

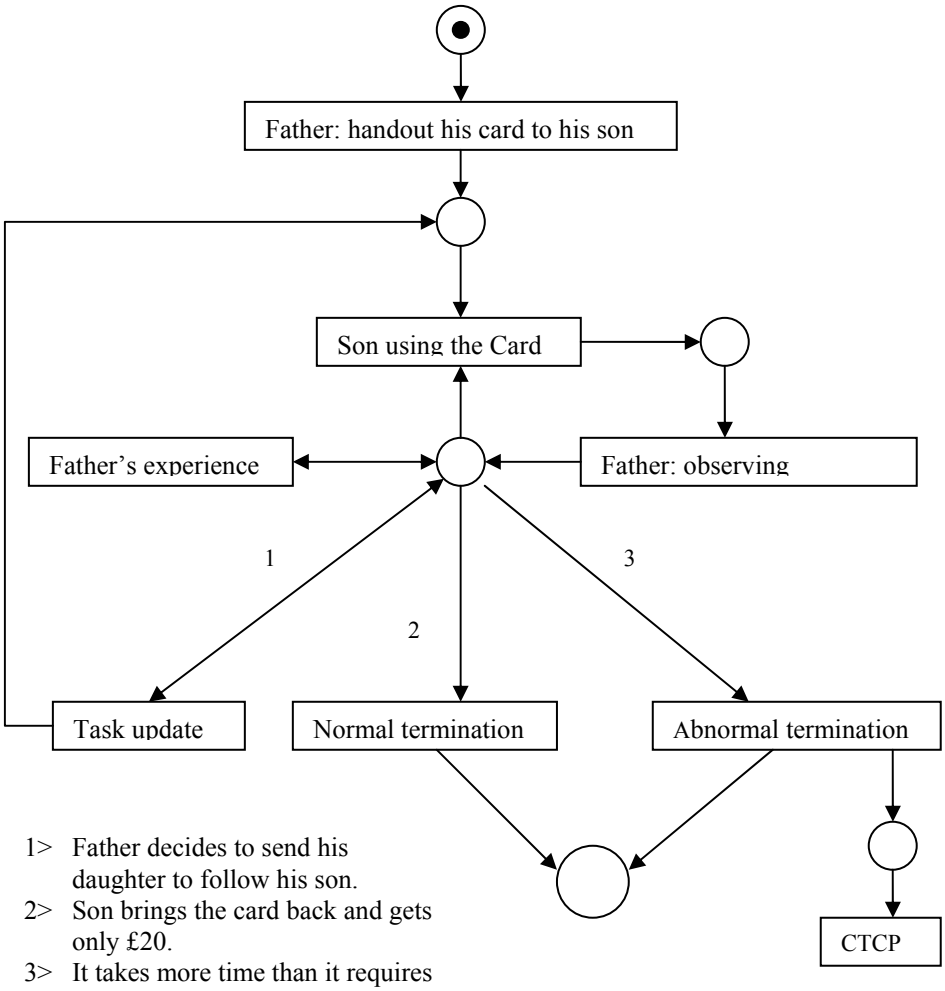

Figure 6: Petri Net representing the CTRP of the given Example

7.3 Example modelling:

In this example we have two collaborators (*task-participant1*=son and *task-paticipant2*=father), *task_name* = borrow credit/debit card, *purpose* = withdraw 20 pounds to buy a book, *policy* = {withdraw only 20 pounds, not to use the card for any other purpose, return the card after getting the money, and not to give it away}, and *security-base* = {trust based on experience}. A Petri Net graph in Figure 5 represents this example in the CTCP layer/stage and Figure 6 represents it in the CTRP layer.

8. DISCUSSION

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

The idea of task-based has been introduced before in a number of models [9],[17], [18]. All were at the basic level of this approach. The focus in [17, 18] was on whether a task-based security model could be an alternative authorisation and access control model to the subject-object traditional authorisation models. While in [9] Fischer-Hubner and Ott tried to address the privacy problem using the task-based approach. We intend 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. In more detail:

1. Steinke [17] outlines the general features and characteristics of the task-based approaches such as:
   - The need-to-know is related to the operation, which needs to be performed.
   - Any information needs can be related to a task.
   - Tasks are common entities that exist and relate directly to both users and to information.
   - Tasks limit the access to the information from the start to the termination of the tasks.
   - Tasks already exist, and are identifiable, flexible and dynamic.

The Group Security model (GSM) by Steinke was described as a security model, which provides access to information on the base of a user's task.

However some features of GSM are already rather obvious in existing information systems infrastructure. For instance in any relational database, it is always possible to grant users/roles to functions, procedures, and packages rather than grant them to the information objects (e.g. tables, views). These functions, procedures and packages are in fact tasks and group of tasks and also can be functionally minimized. GSM considers the discretionary security approach to deal with ownership. Overall GSM is more suitable for hierarchy systems, where the responsibilities are visible.

2. Thomas and Sandhu [18] introduced the task-based approach initially in 1994 as an approach to address integrity issues in computerized information systems from an enterprise perspective. Subsequently in 1997 they [19] developed their approach to produce a paradigm for access control and authorisation management. The developed model is called Task-based authorisation control TBAC.
3. Fischer-Hubner and Ott [9] in their model attempted to address the privacy aspect using the task-based approach. The nature of the task-based approach eases the handling of the main privacy requirements such as:

- Purpose binding: personal data obtained for one purpose should not be used for another purpose without informed consent.
- Necessity of data collection and processing: the collection and processing of personal data shall only be allowed, if it is necessary for tasks falling within the responsibility of the data processing agency.

In contrast to the models of Steinke and of Thomas and Sandhu, this model takes a forward step to de-centralise the authorisation using a 4-eyes principle. However there were no end-user requirements supporting this model and the 4-eyes principle is not enough to ensure de-centralisation. The set theory which was used to represent this model is not proven, nor is it in a framework (Petri nets, Category theory, LaSCO, Ponder, VDM, Z, ...) where proof is done by following constructive principles or through following rules guaranteeing a particular outcome. Finally the Fischer-Hübner and Ott model does not include collaboration ventures.

4. Mahling, Coury and Croft [13], 11 years ago, tried to build a task-based collaboration model. However this work starts from a relatively late stage in the negotiation where the plan, agreement and tasks are relatively clear. In addition their work does not consider the case of the multi-agency environments where the policies of the collaborators are different.

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 have considered the multi-agency aspects in detail. Furthermore the other issue of any computer system design including security system is the *usability* [12]. This issue was ignored in most of the above security models.

Usability and computability are almost equally important issues. It is not sufficient for a computer system to be robust, dependable, and cover all the expected functions (computability). It also has to be accepted by its users, in other words it has to be user-friendly (usability). Social specialists and some groups of information and computer specialists argue that the issue of "usability" has more to do with developments and implementations of computer systems than their computability. Certainly the computer security issue is not an exception from this rule. Indeed usability factors such as politics, organization policy and rules, human behaviours and modes, groups and individuals' interests are very much involved in the design of a secure computer system. For instance a long and difficult security procedure may affect the system availability and/or encourage users to skip some steps of the procedure. In addition to its coverage of the issue of computability, our model will also suitably fulfill the usability requirements since it considers direct participation of the users. For example it is not necessary, in our task-based model, to fully computerize a given task; it depends very much on the result of the negotiation between the involved parties in that task.

9. CONCLUSIONS

This paper has introduced a task-based model to facilitate collaboration in trusted multi-agency networks, after a wide investigation of the exiting security models concerning the multi-agency environment and collaboration 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 in this paper in the form of Petri Nets, are used to represent the permitted states and transitions. An example of informal collaboration is used to illustrate the application of the model. We have also discussed the extent to which task-based approaches have been used before in security systems. In addition to its coverage of the area of computability, our model suitably covers the usability requirements.

10. REFERENCES

1. Aljareh, S. and Rossiter N. A Task-based Security Model to facilitate Collaboration in Trusted Multi-agency Networks. In proceedings of the 2002 symposium on Applied Computing. Madrid, Spain March 11-14 2002, pp 744-749.
2. Aljareh, S. and Rossiter N. Toward security in multi-agency clinical information services. In proceedings of workshop on dependability in healthcare Informatics Edinburgh, March 22-23 2001, pp 33-41.
3. Anderson, R. A security Policy Model for clinical information systems. In Proceedings of the IEEE Symposium on Research in Security and Privacy, Research in Security and Privacy, pp. 30–43. IEEE Computer Society, Technical Committee on Security and Privacy, IEEE Computer Society Press, Oakland, CA, May 1996.
4. Bell, E. D. and LaPadula, L. J. Secure Computer Systems: Mathematical Foundations. Mitre Report ESD-TR-73-278 (Vol. I–III), Mitre Corporation, Bedford, MA, Apr 1974.
5. Biba, K. Integrity Considerations for Secure Computing Systems. Mitre Report MTR-3153, Mitre Corporation, Bedford, MA, 1975.
6. Brewer, D. F. C. and Nash, M. J. The Chinese Wall Security Policy. In 1989 IEEE Symposium on Security and Privacy, pp. 206–214. Oakland, CA, 1989.
7. Chu-Carroll, J., and Carberry, S., Conflict Resolution in Collaborative Planning Dialogues, International Journal of Human-Computer Studies, **53**(6), 2000, pp 969-1015.
8. Clark, D.D. and Wilson, D. R. A Comparison of Commercial and Military Computer Security Policies. In 1987 IEEE Symposium on Security and Privacy, pp. 184–194. Oakland, CA, 1987.
9. Fischer-Hübner, S. and Ott, A. From a Formal Privacy Model to its Implementation. Proceedings of the 21st National Information Systems Security Conference, Arlington, VA, October 5-8, 1998.
10. Gollmann, D. Computer Security. ISBN: 0 471 97844 2, John Wiley and Sons, 1999.
11. Harrison, M.A., Ruzzo, M.L. and Ullman, J.D. Protection in operating systems. Communication of the ACM, 19(8) pp 461-471, August 1976.
12. Kling, R. Information and computer scientists as moral philosopher and social analysts. Computerization and controversy: Value conflicts and social choices. Academic Press Inc, 1991, pp 32-37.
13. Mahling, D.E., B.G. Coury, and W.B. Croft. User Models in Cooperative Task-oriented environment. Proceeding of the 23rd Annual Hawaii IEEE International Conference on System Science, 1990, pp 94-99.
14. Reisig, W. and Rozenberg G. Lectures on Petri Nets: Advances in Petri Nets. Lecture Notes in Computer Science, 1998, 1491.
15. Reisig, Wolfgang. Petri Nets: an introduction. Berlin; New York: Springer-Verlag, 1985.
16. Shen. H. H. and Dewan, P. Access control for collaboration environment CSCW 1992 pp 51-58.
17. Steinke, G. A task-based Approach to Implementing Computer Security. Journal of computer Information Systems, fall 1997, pp 47-54.
18. Thomas, R. K. and Sandhu, R. S. Conceptual Foundation for a Model of Task-Based Authorization. Proceedings of the 7th IEEE Computer Security Foundations Workshop, Franconia, NH, June 1994, pp 66-79.

19. Thomas, R. K. and Sandhu, R. S. Task-based Authorization Controls (TBAC): A Family of Models for Active and Enterprise-oriented Authorization Management. Proceedings of the IFIP WG11.3 Workshop on Database Security, Lake Tahoe, California, August 11-13, 1997
20. Van der Aalst, W. M. P, and Basten, D, Identifying Commonalities and differences in Object Life Cycles using Behavioral Inheritance, Application and Theory of Petri Nets 2001, 22nd International conference ICATPN, Newcastle, 2001, 32-52.