Modelling Trust in the *i** Strategic Actors Framework

Eric Yu and Lin Liu
Faculty of Information Studies, University of Toronto
Toronto, Ontario, Canada M5S 3G6
{yu, liu }@fis.utoronto.ca

ABSTRACT

The ***i**** framework models intentional dependency relationships among strategic actors and their rationales. Actors depend on each other for goals to be achieved, tasks to be performed, and resources to be furnished. The concept of softgoal is used to model quality attributes for which there are no *a priori*, clear-cut criteria for satisfaction, but are judged by actors as being sufficiently met ("satisficed") on a case-by-case basis. The framework was developed to support requirement analysis and high-level design in an agent-oriented system development paradigm. In this paper, we explore the use of ***i**** for modelling trust relationships. Trustworthiness is treated as a softgoal to be satisficed from the viewpoint of each stakeholder. Contributions to trustworthiness are considered using a qualitative reasoning approach. Examples from the smart card domain are used to illustrate.

1. INTRODUCTION

Trust is becoming a central issue in today's increasingly networked information systems. For example, in electronic commerce, exchanges often take place among parties unfamiliar to each other, and whose identities may be transitory. Different parts of networks and systems may be operated by parties with conflicting interests or even malicious intent. Furthermore, many of the technologies, as well as business models, are new and their viability is unproven.

Techniques for systems analysis and design have, in the past, focused primarily on addressing functional requirements, assuming that all parties are trusted. Given today's environments, there is need for new techniques that would bring issues of trust, risk, and vulnerability prominently into the system analysis and design process.

The ***i**** framework [12] was developed for modelling intentional relationships among strategic actors. Actors have freedom of action, but operate within a network of social relationships. Specifically, they depend on each other for goals to be achieved, tasks to be performed, and resources to be furnished. These dependencies are intentional in that they are based on underlying concepts such as goal, ability, commitment, belief, and so on. Actors are strategic in that they evaluate their social relationships in terms of opportunities that they offer, and vulnerabilities that they may bring. Strategic actors seek to protect or further their interests. Compared to conventional modelling techniques such as data flow diagramming and object-oriented analysis (e.g., UML), ***i**** provides a higher level of modelling so that one can reason about opportunities and vulnerabilities. The framework has been elaborated in the context of requirements engineering [13], business processing reengineering [16] [14], and software processes [15]. The framework is being extended to form the basis of an agent-oriented system development paradigm.

In this paper, we explore the use of ***i**** for modelling trust relationships. Trustworthiness is treated as a softgoal to be satisficed from the viewpoint of each stakeholder. The concept of softgoal is used to model quality attributes for which there are no *a priori*, clear-cut criteria for satisfaction, but are judged by actors as being sufficiently met ("satisficed") on a case-by-case basis. Contributions to trustworthiness are systematically elaborated and analyzed using a qualitative reasoning approach. The softgoal concept in ***i**** arose from an approach to dealing with non-functional requirements in software engineering. Non-functional qualities of a system have to do not with the functions that the system provides, but how well they are accomplished, e.g., how speedily (performance), how cheaply (costs), how accurately, etc. Many non-functional requirements are hard to quantify or characterize, e.g., flexibility, maintainability, evolvability, scalability, etc. An important feature of these non-functional qualities is that they interact with each other in complex ways. The NFR framework [3] [4] offers a graphical notation and framework for systematically elaborating and analyzing the contribution relationships in a network of softgoals. Contributions can be positive and negative, and may be considered partial or sufficient towards addressing some softgoal. The ***i**** framework interleaves non-functional analysis with the functional analysis of system operation. These are done within a network of social actors. Actors may be further differentiated into agents, roles, and positions.

An example from the smart card domain is used to illustrate. Only a subset of the features of ***i**** are illustrated in this paper. Section 2 presents an overview of the ***i**** framework, introducing its basic concepts using the smart card example. Section 3 considers the modelling of attacks and defense, first from each attacker's viewpoint, then combined with defender's countermeasures. An outline of the qualitative evaluation method for propagating satisificing judgements across the network model is also provided. Section 4 discusses related work.

2. AN OVERVIEW OF THE *i** FRAMEWORK

The framework includes a Strategic Dependency model – for describing the network of relationships among actors, and a Strategic Rationale model – for describing and supporting the reasoning that each actor has about its relationships with other actors.

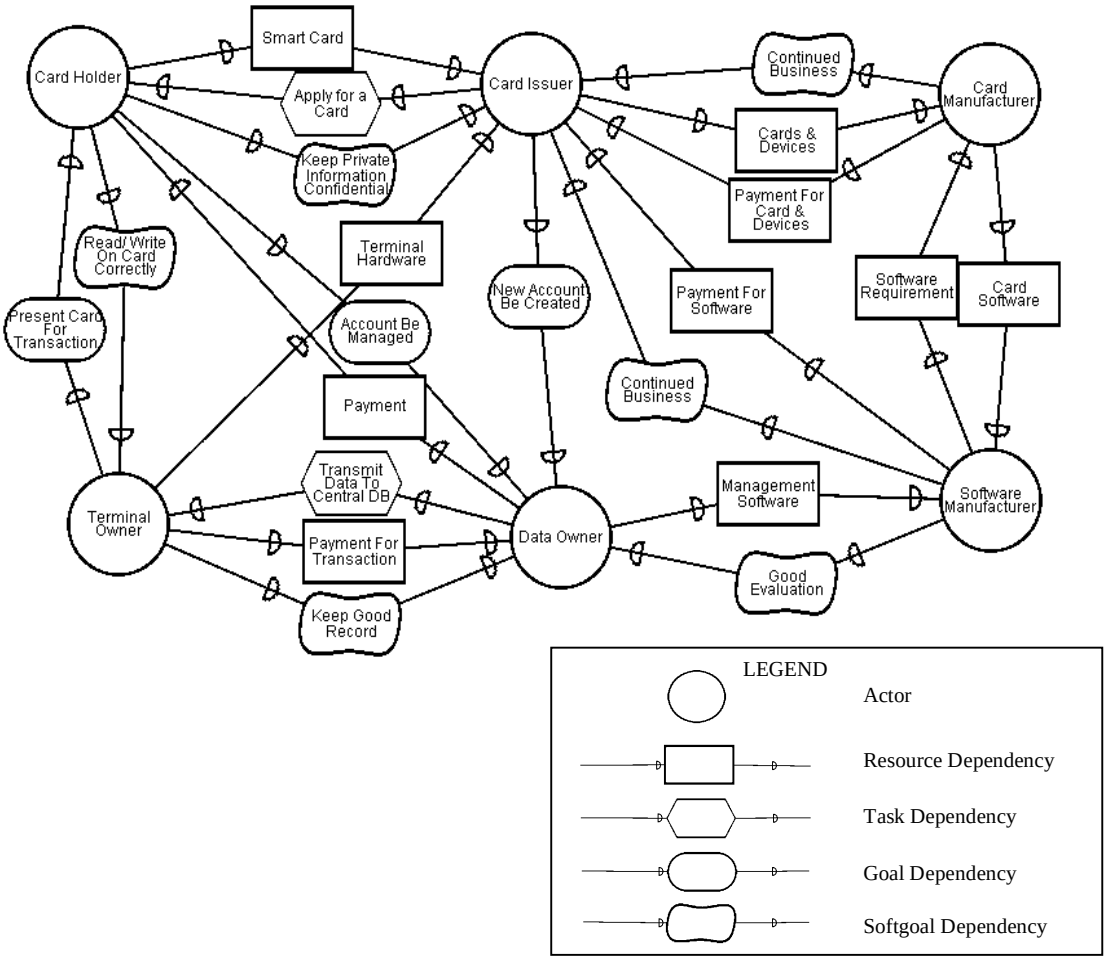

Figure1: Strategic Dependency model of smart card based payment system

2.1 The Basic Strategic Dependency Model

A Strategic Dependency (SD) model consists of a set of nodes and links. Each node represents an actor, and each link between two actors indicates that one actor depends on the other for something in order that the former may attain some goal. We call the depending actor the *depender*, and the actor who is depended upon the *dependee*. The object around which the dependency relationship centres is called the *dependum*. By depending on another actor for a dependum, an actor (the depender) is *able* to achieve goals that it was not able to without the dependency, or not as easily or as well. At the same time, the depender becomes vulnerable. If the dependee fails to deliver the dependum, the depender would be adversely affected in its ability to achieve its goals.

Figure 1 shows a Strategic Dependency model for a generic smart card-based payment system. A cardholder depends on a card issuer to be allocated a smart card, for the terminal depends on him to present his card for each transaction. The card issuer in turn depends on the card manufacturer and software manufacturer to provide cards, devices, and software. The data owner is the one who has control of the data within the card. He depends on the terminal to submit transaction information to the central database.

The Strategic Dependency model distinguishes among several types of dependencies, based on the ontological category of the dependum. In a *goal dependency*, an actor depends on another to make a condition in the world come true. Because only an end state or outcome is specified, the dependee is given the freedom to choose how to achieve it. In the example of Figure 1, the goal dependency "new account be created" from the card issuer to the data owner means that it is up to the data owner to decide how to create a new account. The card issuer does not care how a new account is created, what matters is that, for each card, an account should be created.

In a *task dependency*, an actor depends on another to perform an activity. The depender's goal for having the activity performed is not given. The activity description specifies a particular course of action. The card issuer depends on the cardholder to apply for a

card via a task dependency by specifying standard application procedures. If the card issuer were to indicate the steps for the data owner to create a new account, then the data owner would be related to the card issuer by a task dependency.

In a *resource dependency*, an actor depends on another for the availability of an entity. The depender takes the availability of the resource to be unproblematic. In Figure 1, the card issuer's dependencies on the card manufacturer for cards and devices, the manufacturers' dependencies on card issuer for payment are modelled as resource dependencies.

The fourth type of dependency, *softgoal dependency*, is a variant of the first. It is different from a (hard) goal dependency in that there is no *a priori*, cut-and-dry criteria for what constitutes meeting the goal. The meaning of a softgoal is specified in terms of the methods that are chosen in the course of pursuing the goal. The dependee contributes to the identification of alternatives, but the decision is taken by the depender. The notion of the softgoal allows the model to deal with many of the usually informal concepts. For example, the manufacturers' dependencies on the card issuer for continued business can be achieved in different ways. The desired style of continued business is ultimately decided by the depender. The cardholder's softgoal dependency on the card issuer for "keep private information confidential" indicates that there is not a clear-cut criterion for the achievement of confidentiality. The four types of dependencies reflect different types of freedom that is allowed in the relationship between depender and dependee.

The Strategic Dependency model of Figure 1 is not meant to be a complete and accurate description of any particular smart card system. It is intended only for illustrating the features of ***i****.

2.2 Roles, Positions, and Agents

In ***i****, the term *actor* is used to refer generically to any unit to which intentional dependencies can be ascribed. To model complex relationships among social actors, we further define the concepts of agents, roles, and positions, each of which is an actor in a more specialized sense. A basic Strategic Dependency model can be extended by refining the notion of actor into notions of role, position, and agent.

An *agent* is an actor with concrete, physical manifestations, such as a human individual. An agent has dependencies that apply regardless of what role he/she/it happens to be playing. For example, if Jim, a cardholder desires a good credit record, he actually wants the credit record to go towards his personal self, not to the positions and abstract roles that Jim might occupy or play. We use the term of agent instead of person for generality, so that it can be used to refer to human as well as artificial (hardware, software, or organizational) agents. In Figure 2, customer and merchant are represented as agents.

A *role* is an abstract characterization of the behavior of a social actor within some specialized context or domain of endeavor. Dependencies are associated with a role when these dependencies apply regardless of who plays the role. For example, we consider attacker and defender as two roles any actor can play. No matter who plays the role of attacker, he will have a high level goal of

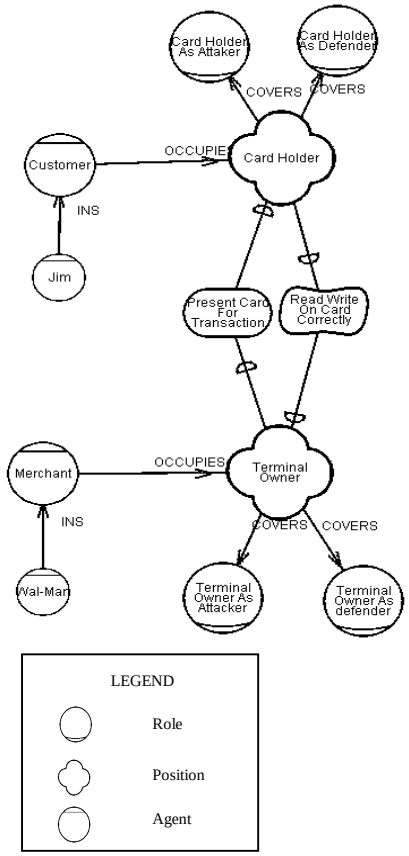

Figure 2: Strategic Dependency model with roles, positions, and agents

"attack". Regardless of who plays the role of defender, he will have the goal of "defense".

A *position* is intermediate in abstraction between a role and an agent. It is a set of roles typically played by one agent. Positions can *cover* roles, agents can *occupy* positions, and agents can also *play* roles directly.

Figure 2 shows a fragment of the Strategic Dependency model from the smart card example with agents, roles, and positions. In this partial model, a cardholder position covers two roles of cardholder as attacker and cardholder as defender. The position of cardholder is occupied by the agent "customer".

The "INS" construct represents the instance-and-class relation. For example, Wal-Mart is an instance of merchant, and Jim is an instance of customer. The "ISA" construct expresses conceptual generalization/ specialization. For example, a bank is a kind of financial institution. These constructs are used to simplify the presentation of strategic model with roles, positions, and agents.

There can be dependencies from an agent to the position it occupies. For example, a merchant who occupies the position of terminal owner depends on that position to attract more customers. Otherwise, he may choose not to occupy that position.

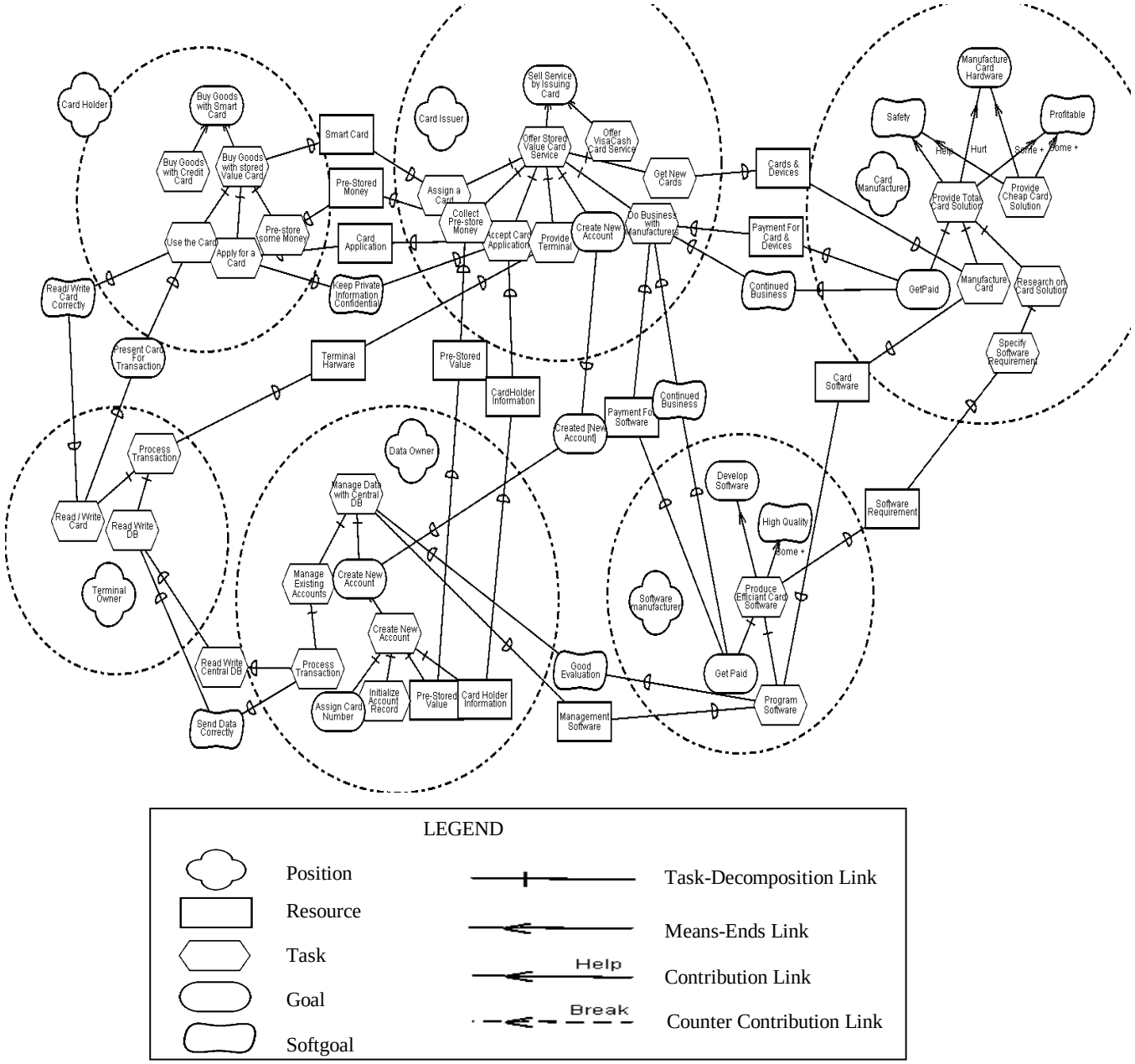

Figure 3: Strategic Rationale model of smart card based system

Roles, positions, and agents can each have subparts. Aggregate actors are not compositional with respect to intention. Each actor, regardless of whether it has parts, or is part of a larger whole, is taken to be intentional. Each actor has inherent freedom and is therefore ultimately unpredictable. There can be intentional dependencies between the whole and its parts, e.g., a dependency by the whole on its parts to maintain unity.

2.3 The Strategic Rationale Model

The Strategic Rationale (SR) model provides a more detailed level of modelling by looking "inside" actors to model internal intentional relationships. Intentional elements (goals, tasks, resources, and softgoals) appear in SR models not only as external dependencies, but also as internal elements arranged into a hierarchy of means-ends and task-decompositions relationships. The SR model in Figure 3 elaborates on the relationships among cardholder, card issuer, data owner, terminal owner, card manufacturer, and software manufacturer as depicted in the SD model of Figure1.

For example, each cardholder has an internal goal of "Buy Goods with Smart Card". When an element is expressed as a goal, it means there might be several alternatives to accomplish this, i.e., the cardholder can either "Buy Goods with Credit Card", or " Buy Goods with Stored Value Card". These are represented as tasks. A

task specifies one particular way of doing things in terms of further decomposition into subtasks, subgoals, resources, and softgoals. Here the task of "Buy Goods with Stored Value Card" is composed of three subtasks: "Apply for a Card", "Pre-store some Money", and "Use the Card".

For a card issuer, "Sell Service by Issuing Card" is his high level goal. This goal can be achieved by issuing different kinds of cards, e.g., stored value card, VisaCash card, prepaid phone card etc. Offering stored value card service involves doing "Accept Card Application", "Collect Pre-Stored Money", "Get a new Card", "Create a new Account", and "Assign a card" to the applicant. At the same time, he also has to "Provide Terminal" to terminal owner. "Create a new account" is a subgoal, indicating that there are different ways to achieve it.

On the side of the card manufacturer, "Manufacture Card Hardware" is his high level goal. One of the two ways to accomplish the goal is to "Provide Total Card Solution" (such as the Mondex solution [10]), the other is "Provide Cheap Card Solution" (such as Millicent Solution [10]). Both solutions contribute somewhat positively to the softgoal "profitable". "Provide Total Card Solution" will help the safety of the system, while "Provide Cheap Card Solution" will hurt the safety of the system.

The positive contribution types for softgoal are Help (positive but not by itself sufficient to meet the higher goal), Make (positive & sufficient) and Some+ (positive in unknown degree). The corresponding negative types are Hurt, Break and Some-. And means if all subgoals are met, then the higher goal will be sufficiently met. Or means the higher goal will be sufficiently met if any of its subgoals are met. During system analysis and design, softgoals such as profitability and safety are systematically refined until they can be operationalized and implemented [6]. Unlike functional goals, nonfunctional qualities represented as softgoals frequently interact or interfere with each other, so the graph of contributions is usually not a strict tree structure [4].

3. STUDY OF TRUST IN SMART CARD BASED SYSTEM

We now use ***i**** to model some aspects of trust in the smart card example. Most of the dependencies in Figure 1 relate to the normal operations of a smart card. In considering potential problems and threats, further dependencies need to be identified (Figure 4). For example, the cardholder depends on the card issuer to provide a card that is usable (as opposed to a fake or defective one). The cardholder also expects the issuer to protect the privacy of the personal information. Note that these are constituent elements that eventually contribute towards the cardholder trusting the card issuer for the operation of the smart card system.

3.1 Analyzing Possible Attacks

If the card issuer is not operating in good faith, the cardholder's expectations may not be met, i.e., the dependencies may not be viable. In the Strategic Rationale model, we model attacks (Figure 5) as negative contributions from the attackers (from their specific methods of attack) toward the dependums. A Break contribution indicates that the attack is sufficient to make the dependum unviable. For clarity of analysis, we place the attack-related intentional elements of the card issuer into a role called "Card Issuer As an Attacker". Details of the attack methods (e.g., privacy invasion, sell unusable card) can be elaborated by further decomposition and means-ends analysis. Negative contribution links can then be used to show attacks on more specific vulnerability of the depender (e.g., refinements of "Privacy Be

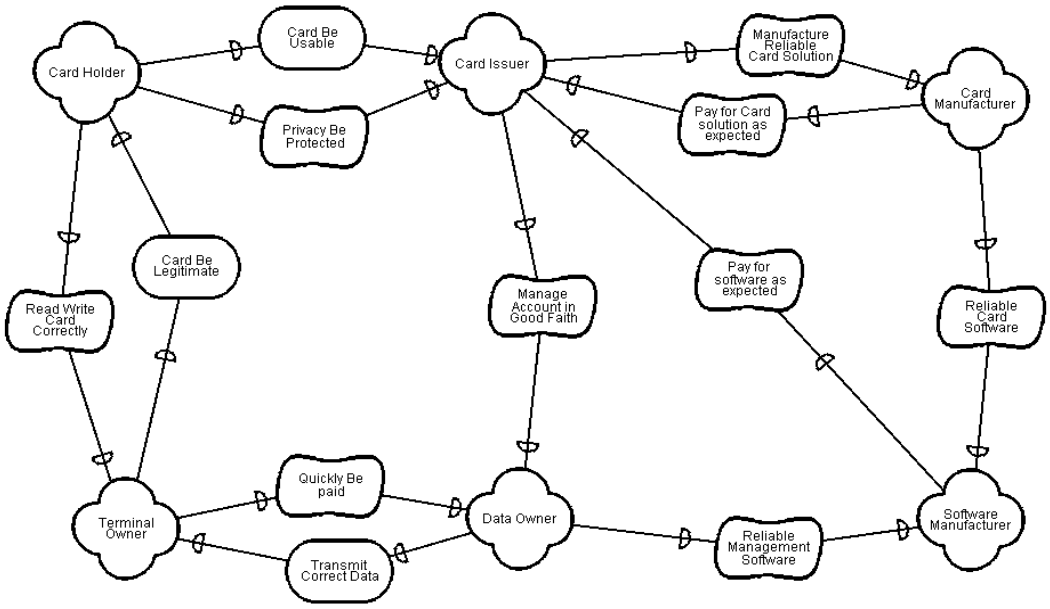

Figure 4: A Strategic Dependency model depicting some trust-related relationships in a smart card system

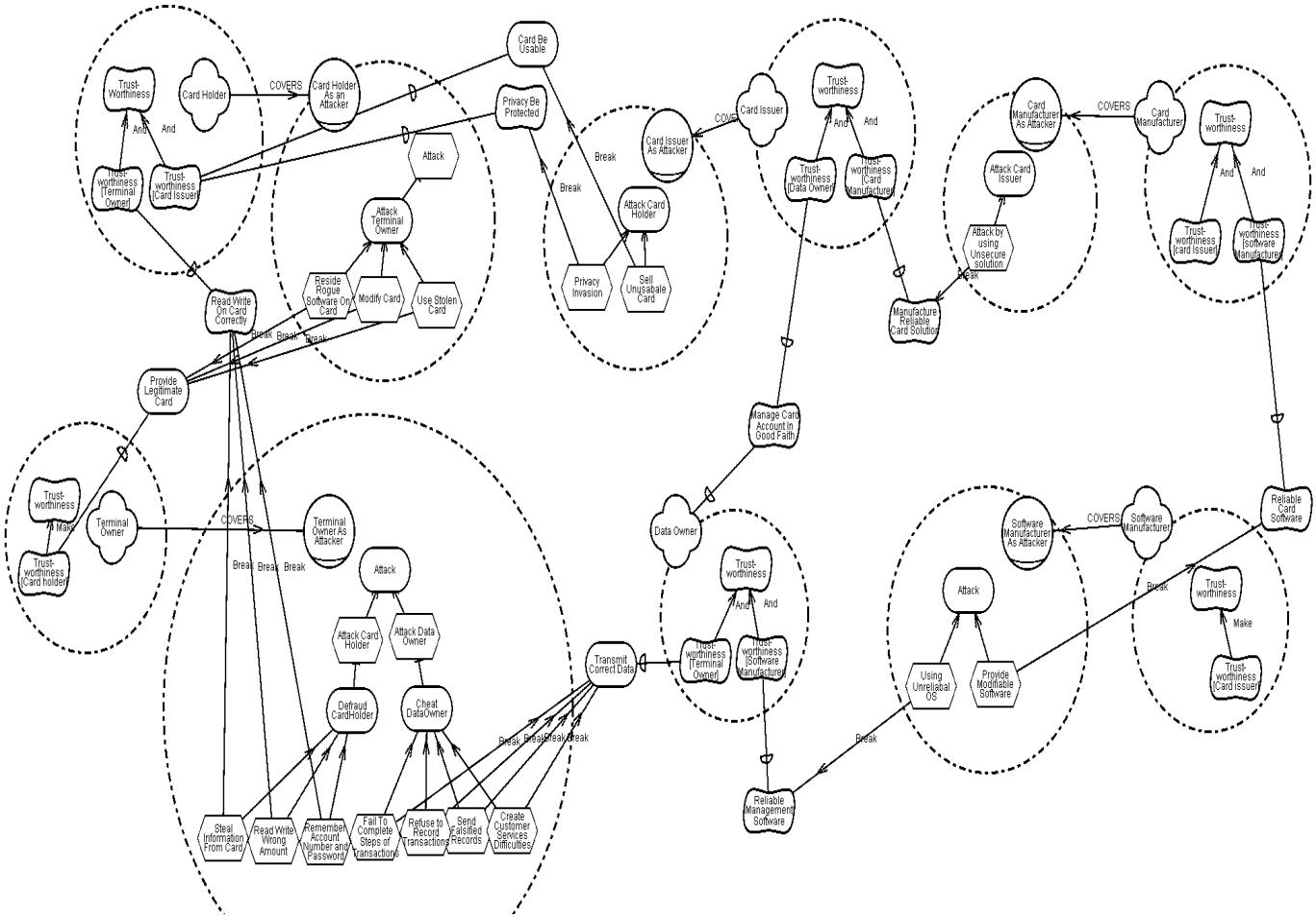

Figure 5: A Strategic Rationale model showing some rationales in attacker roles

Protected"). The refinements (and possible attack routes) may be based on analysis of the SD and SR models of the normal operations of the smart card, e.g., what resources an actor accesses, what types of interactions exist, etc.

For the cardholder to trust the smart card system, he has to trust both the card issuer And the terminal owner. The cardholder depends on the terminal owner to "Read/Write Card Correctly". If the terminal owner is malicious (Terminal Owner As Attacker), there are a number of attacks that are sufficient to make that dependency not viable (Break). Note that each part may potentially attack any other part that it interacts with.

3.2 Modelling Defensive Actions

With the knowledge of some possible attacks, actors may change their methods of operation, or add countermeasures to protect their interests and security. Figure 6 shows a SR model with defender roles as well as attacker roles. Defense mechanisms are adopted to counteract specific attacks. In some cases, defenses can be found which are sufficient to defeat a strong attack (defense Break link (dotted arrow) pointing to an attack Break link). For example, each of "Use Monitors on Back End" and "Make Secure Connection between Card and Back End" is considered to be sufficient for overcoming the four different attacks from the terminal owner to the data owner's dependency of "Transmit Correct Data".

Other countermeasures may only be partially effective in defending against their respective attacks (through the Hurt or Some- contribution types).

Unviable dependencies due to potential attacks lead to erosion of trust of the smart card system. Incorporating sufficient countermeasures restores trust.

3.3 Evaluating System Trust Situation

Having created the model with attacks as well as countermeasures against them, we can evaluate the trust situation under the current system configuration.

Figure 7 shows some of the evaluation of the model of Figure 6. The process of evaluation used is an interactive labeling algorithm, which propagates a series of labels through the modelling framework [4]. A label (or satisficing status) on a node is used to indicate whether that intentional element (goal, task, resource, or

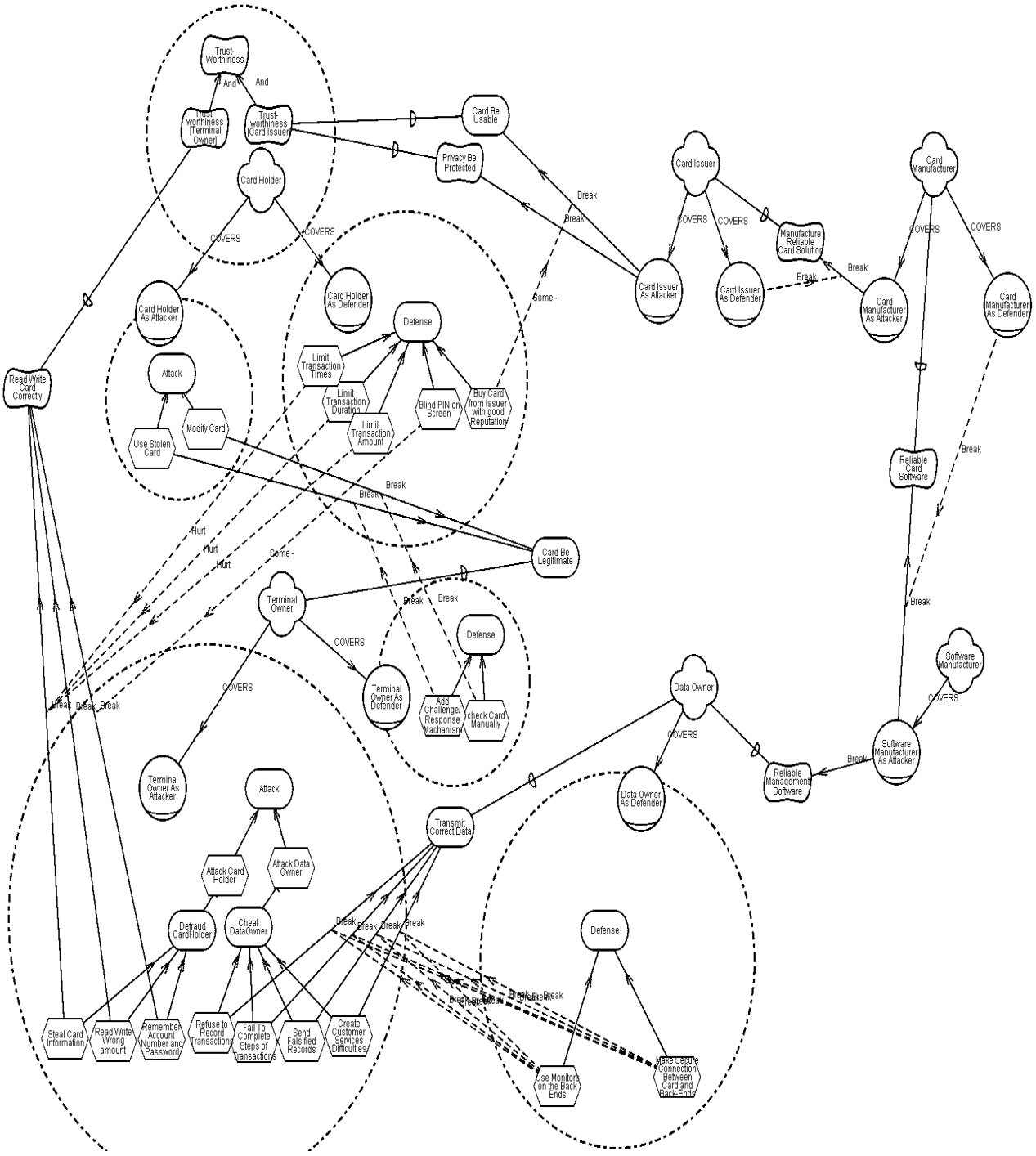

Figure 6: A Strategic Rationale model showing details of selected attacker roles and defender roles

softgoal) is viable or not (e.g., whether a softgoal is sufficiently met). A qualitative reasoning scheme is used. Labels can have values such as Satisfied (√), Denied (⨂), Weakly Satisfied (W+) and Weakly Denied (W-), Conflict, etc. Leaf nodes (those with no incoming contributions) are given labels by the analyst based on judgement of their independent viability. These values are then propagated "upwards" through the contribution network. The viability of the overall system appears in the high level nodes of the various stakeholders. The propagation procedure is described in [4]. It is an interactive one requiring the analyst to make judgements whenever the outcome is inconclusive given the combination of incoming contributions.

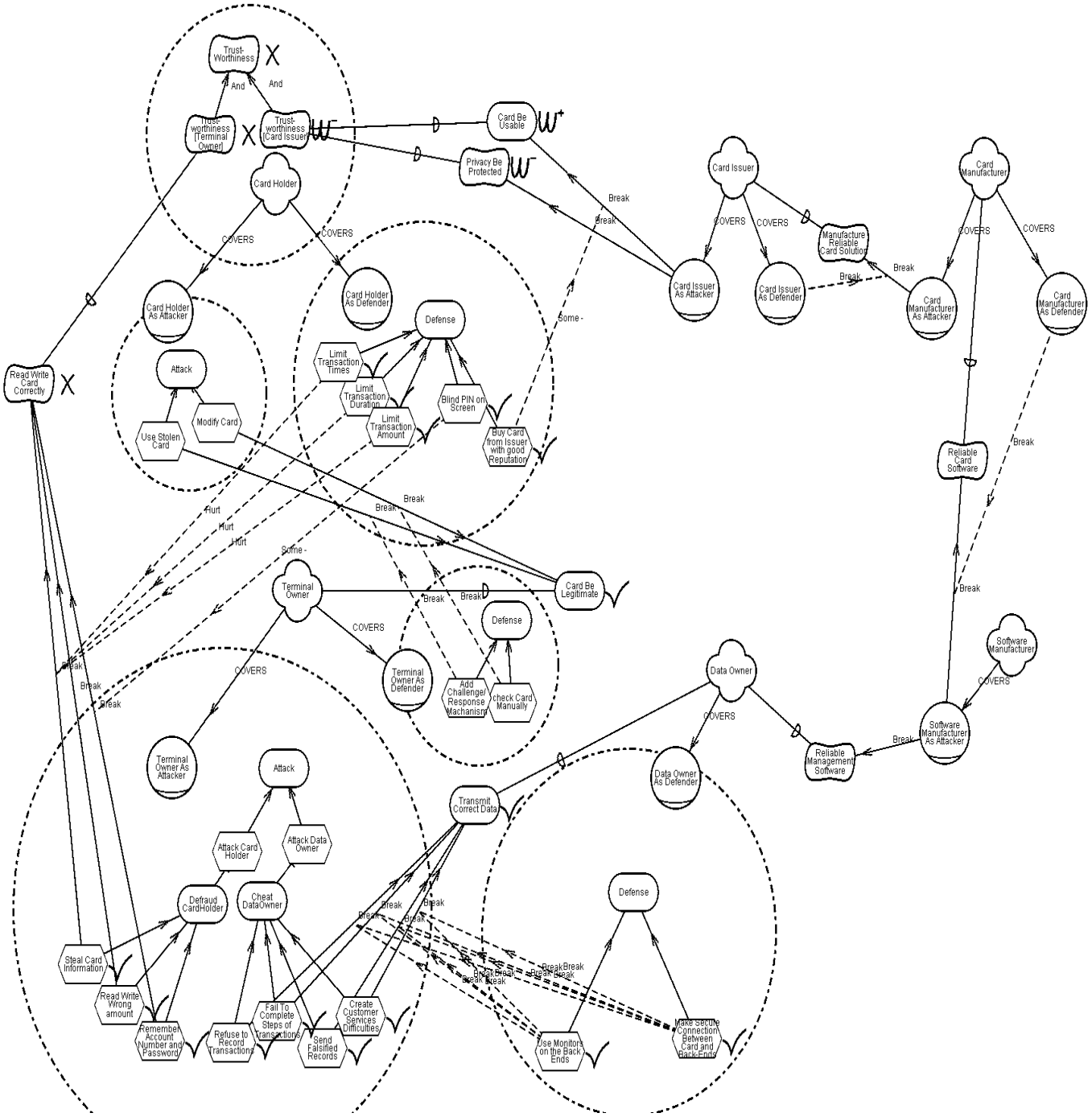

Figure 7: A labeled Strategic Rationale model showing evaluation details of selected attacker and defender roles

In the example of Figure 7, the analyst labels all the attack leaf nodes as "Satisficed" since they are all judged to be possible. Similarly all the defense leaf nodes are judged to be viable, thus labelled "Satisificed". The values are then propagated along contribution links.

Consider the cardholder's situation. He has dependencies on the card issuer and terminal owner. He has a defense "Buy Card From Issuer With Good Reputation" against the card issuer's attack on "Card Be Usable". But the defense may only be a partial one (Some-). So the dependency is judged to be weakly satisficed (W+). The attack by the card issuer on "Privacy Be Protected" is a partial one (Hurt), but the Cardholder has no defense for it. So the dependency is judged to be weakly denied (W-). The combination of these two contributions leads to the judgement that the trustworthiness of the card issuer is weakly denied (W-).

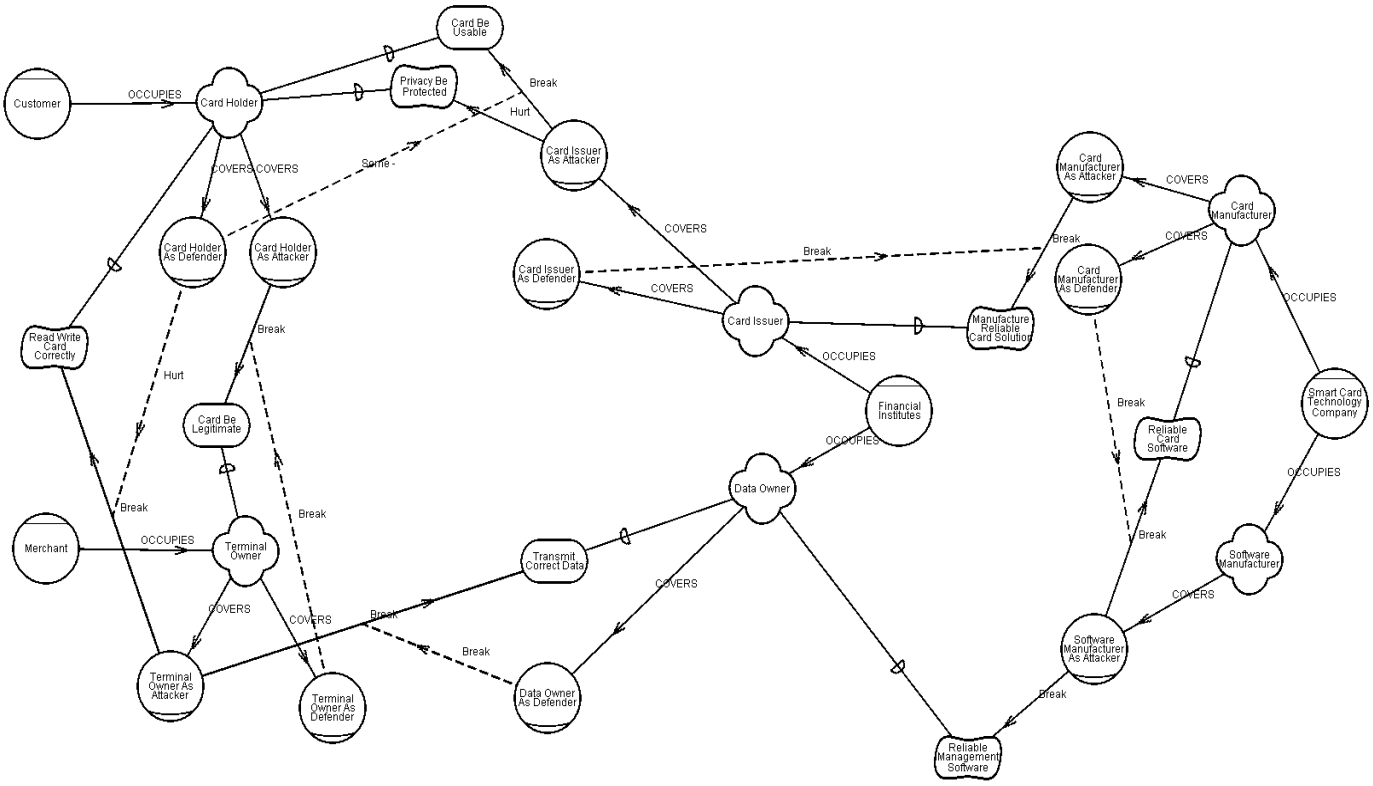

Figure 8: A Strategic Dependency model of a store value smart card system

Regarding the cardholder's dependency on the terminal owner for "Read/Write Card Correctly", there are three possible attacks. One of them "Steal Card Info" is counteracted by three defense measures, though each one is partial (Hurt). Another attack "Remember Account Number & Password" has a defense of unknown strength (Some-). The third attack has no defensive - measure. The "Read/Write Card Correctly" dependency is thus judged to be unviable. The Trustworthiness of Terminal Owner is denied, leading to the Cardholder to conclude that its Trustworthiness goal of the smart card system is not met.

When all known ways in which a system can be attacked are resisted by countermeasures that are strong enough, the system may be judged to be a trustworthy system. The model needs to be revised as new attack routes are identified, and when countermeasures are installed. The reasoning in the model can further be justified by a network of beliefs or assumptions. These are not shown in this paper. Having created the model with attacks as well as their countermeasures, we can further evaluate the trust situation under various system configurations.

3.4 Dealing with Changes of Configuration

In the above modelling, the various participants in a smart card system were modelled as positions and analyzed generally. However, in real world smart card systems, specific organizational parties occupy these positions. Thus, to actually understand their trust situations, we have to apply the generic model to the real world configurations. We consider two representative kind of smart card based systems. One is the Digital Stored Value Card, the other is the Prepaid Phone Card [11].

3.4.1 Digital stored value cards

These are payment cards intended to be substitutes for cash. Both Mondex and VisaCash are examples of this type of system. The cardholder is the customer. The terminal owner is the merchant. The data owner and the card issuer are both the financial institutions that support the system. The card manufacturer and software manufacturer are both technology providers like Mondex.

In such a configuration, the previously isolated positions of data owner and card issuer are occupied by the same physical agent, namely, Financial Institution. Similarly, card manufacturer and software manufacturer are combined into one physical agent – the Smart Card Technology Provider. Figure 8 describes the Strategic Dependency model of a digital stored value card. Here the software manufacturer's attack to card manufacturer can be ignored since they belong to the same agent – the smart card technology company.

3.4.2 Prepaid phone cards

These are simply special-use stored value cards. The cardholder is the customer. The terminal owner, data owner, manufacturer and card issuer are all combined into one agent – the phone company. Figure 9 shows the Strategic Dependency model of a prepaid card system. Under such a system configuration, more attack-defense

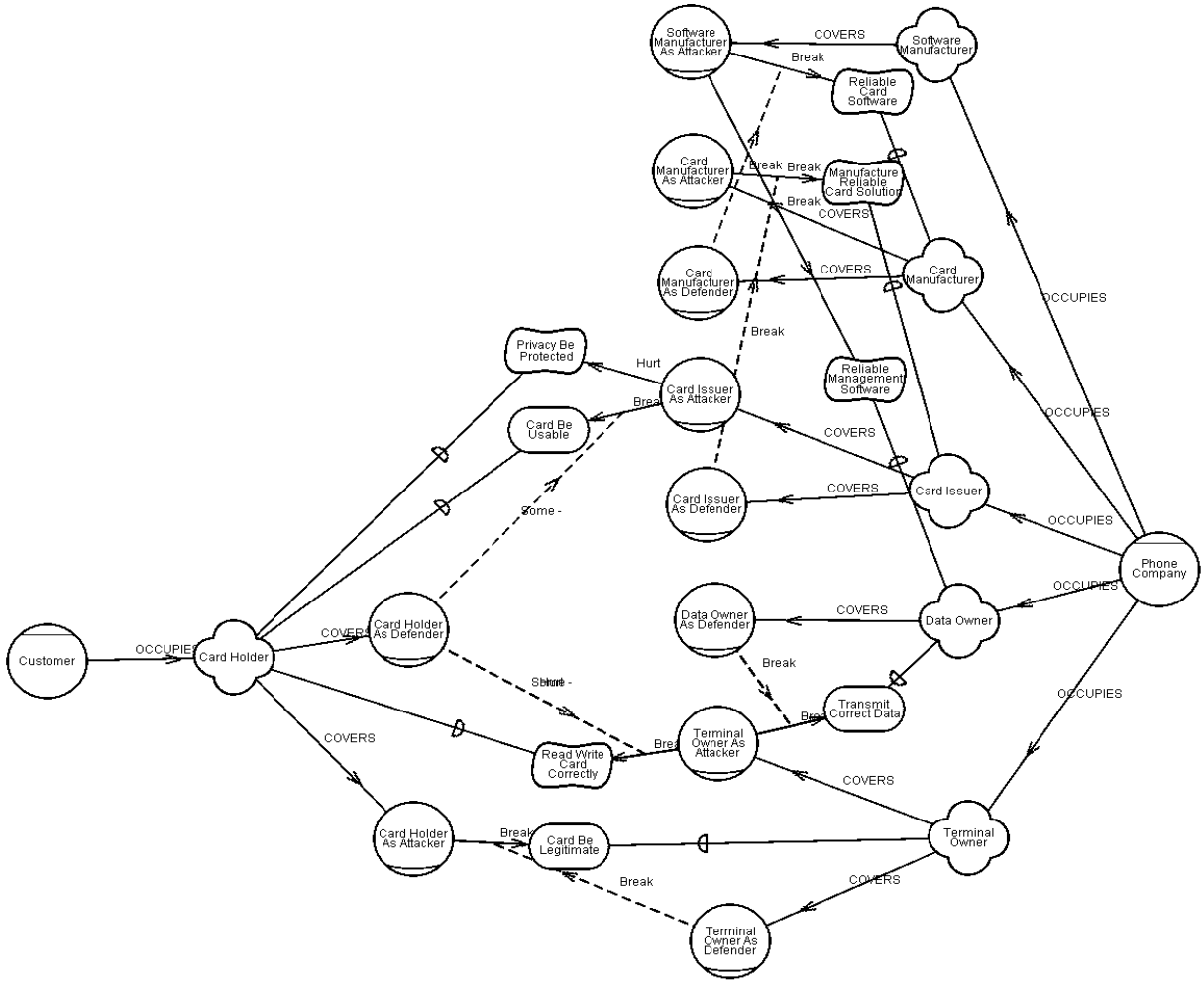

Figure 9: A Strategic Dependency model of prepaid phone card system

pairs disappear. Only four possible attacks need to be considered now. Three of them are from the phone company, which are to hurt privacy, to issue unusable card, to read write card incorrectly. The other attack is from the cardholder, who might use an illegitimate card.

Note that each time new positions are created, the possibility of new attacks arises. These models reflect Schneier's observation that the fewer splits we make, the more trustable the target system might be [11].

4. DISCUSSION

In this paper, we have illustrated the use of ***i**** to model some sample smart card environments. Note that trust is not treated as a distinguished concept with special semantics. Instead, trust is modeled as one of the goals that strategic agents pursue, among others. Thus, it competes with, or complements, as the case may be, other goals such as convenience, profitability, performance, time-to-market, etc.

This approach encourages and facilitates the analysis of trust-related issues within the full operational and social context of the involved actors. The models can be used to encompass normal-case operational procedures, potential attacks, countermeasures against perceived threats, as well as factors not directly related to trust or security. For example, trust issues can be examined in the context of customers' options among different kinds of payment systems, e.g., credit card, cash, smart card, with tradeoffs among convenience, security, and privacy issues.

This approach is complementary to the various theories and techniques currently being developed for specifically addressing trust (e.g., [1] [2]). ***i**** offers a structural representation of intentional relationships among actors and within actors, as well as structural concepts such as intentional agents, roles, and positions. These may provide a structural framework for integrating other concepts and techniques for dealing with trust. For example, the qualitative reasoning approach of ***i**** may be used in a first-pass preliminary analysis, to be followed by techniques with stronger semantics.

The approach is also consistent with the recent call for "reinventing" security [9]. The ***i**** approach acknowledges vulnerability (and thus insecurity) as an inherent feature of any social network, because of the intentional and strategic nature of dependencies. In general, vulnerabilities cannot be totally eliminated, so the emphasis is on finding measures that are sufficiently strong to mitigate them. Also insecurity can be moved around, as demonstrated by the different allocations of positions to agents in the smart card example.

A related area of work is the use of threat trees or attack trees (e.g., [15], [8]) in security analysis. Attack trees describe all possible attacks against a system in a tree structure. The central idea is to use goal-decomposition as in AI. Different ways of achieving an attack are explored and evaluated in terms of possibility and the presumed cost. The ***i**** approach draws on similar ideas but embeds intentional and means-ends reasoning into a network of social actors with dependencies. The model may include normal operations, attacks and defenses among all parties, not just attacks from one viewpoint. A notion of satisficing within a qualitative reasoning framework is used to capture the application of judgemental thresholds at each decision point in a reasoning network.

This paper has taken a rather simplistic view of the nature of trust, for example, by making a fairly direct connection to security in the softgoal graphs. In future work, we plan to incorporate more detailed analysis of the varieties of trust being identified in the emerging literature in this area. We also hope to use ***i**** to analyze broader issues related to trust, including issues of privacy, power, and public policy, as discussed, for example in [7]. A tool is currently being developed to support modelling and reasoning using the ***i**** framework.

REFERENCES

[1] *Autonomous Agents '98 Workshop Proceedings on "Deception, Fraud and Trust in Agent Societies"*, Minneapolis/St Paul, USA, May 9, 1998.

[2] C. Castelfranchi, Y.-H. Tan, R. Falcone, and B. S. Firozabadi, eds. *Autonomous Agents '99 Workshop Proceedings on "Deception Fraud and Trust in Agent Societies"*, Seattle, Washington. May 1999.

[3] L. Chung, *Representing and Using Non-Functional Requirements for the Information System Development: A Process-Oriented Approach*, Ph.D. Thesis, also Tech. Report DKBS-TR-93-1, Dept. of Comp. Sci. University of Toronto, June 1993.

[4] L. Chung, B.A. Nixon, E. Yu, and J. Mylopoulos, *Non-Functional Requirements in Software Engineering*. Kluwer Academic Publishers, 2000.

[5] J. Mylopoulos, L. Chung, B. Nixon, Representing and Using the Non-Functional Requirements: A Process-Oriented Approach, *IEEE Trans. Soft. Eng.* 18(6), June 1992.

[6] J. Mylopoulos, L. Chung, and E. Yu, From Object-Oriented to Goal-Oriented Requirements Analysis, *Communications of the ACM*, 42(1): 31-37, January 1999.

[7] F. Stalder, A. Clement, Exploring Policy Issues of Electronic Cash: The Mondex Case, *Canadian Journal of Communication*, 24(2). 1999.

[8] S. Salter, O. Saydjari, B. Schneier, and J. Wallner, Toward a Secure System Engineering Methodology, *New Security Paradigms Workshop 1998 Proceedings*, IEEE Computer Society Press.

[9] Fred B. Schneider, ed. *Trust in cyberspace*. Committee on Information Systems Trustworthiness, Computer Science and Telecommunications Board, National Research Council. Washington, D.C.: National Academy Press, 1999. Also available at http://cryptome.org/tic.htm.

[10] B.Schneier, Attack Trees Modelling Security Threats. *Dr.Dobb's Journal*, December 1999. Also available at http://www.counterpane.com/attacktrees-ddj-ft.html.

[11] B. Schneier, A. Shostack, Breaking Up Is Hard To Do: Modelling Security Threats for Smart Cards. Available at http://www.counterpane.com/smart-card-threats.html. Also *First USENIX Symposium on Smart Cards*, USENIX Press, to appear.

[12] E. Yu, *Modelling Strategic Relationships for Process Reengineering*, Ph.D. thesis, also Tech. Report DKBS-TR-94-6, Dept. of Computer Science, University of Toronto, 1995.

[13] E. Yu, Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering, *Proc. 3rd IEEE Int. Symp. On Requirements Engineering (RE'97)*, Annapolis, Maryland, USA, January 1997.

[14] E. Yu, J. Mylopoulos, From E-R to 'A-R' – Modelling Strategic Relationships for Business Process Reengineering, *Int. Journal of Intelligent and Cooperative Information Systems*, 4(2&3), 1995, pp.125-144.

[15] E. Yu, J. Mylopoulos, Understanding 'Why' in Software Process Modelling, Analysis, and Design, *Proc. 16th Int. Conf. On Software Engineering*, May 1994, pp. 159-168.

[16] E. Yu, J. Mylopoulos, and Y. Lespérance, AI Models for Business Process Reengineering, *IEEE Expert*, August 1996, pp.16-23.