Requirements for a Human-Centric Trust Management System

in an Open De-Perimeterised Network Environment

Doctoraal (Graduation) Thesis

Department of Information and Computing Sciences
Utrecht University, The Netherlands
Andor Demarteau

August 2008

Abstract

In an ever changing business environment where online collaboration between individuals and companies becomes more and more important, the way we used to think about network security is rapidly becoming obsolete and insufficient. The current practise revolves around perimeter protection with firewalls and access control lists for network and system security and access control. Companies deal with the new ways of online communication by creating openings in the current perimeter to allow them to pass through the security barriers, often without any validation or other security mechanisms in place. This is how security holes are created in the current network design, introducing vulnerabilities in the current system. Trying to fix this with "Simple" introducing detection systems will not be enough. Instead a radical change in thinking about security is required to solve these issues. The Jericho Forum [1], which is part of the Open Group [2], proposes a new way of thinking about security, based on user-centric authentication and access-controls placed on access to data rather than access to networks and systems.

In this paper we will look at the trust broker part within the Jericho Architecture. More specifically, the trust management system, which is the basis of access-control within Jericho and is based upon identity recognition, trust relationships based on recommendation and observation. Before we can define trust and reputation in a technical context, we need some method of describing them and the meaning they have in the human-centric real world context of everyday life. For this purpose we will use Perceptual Control Theory (PCT). This theory, based in the social sciences, gives a description of how human behaviour works by using closed causal loops of control. These loops are part of an entire hierarchy which defines a complete "living" control system. As our definitions of human trust relationships will be based on this theory, we also will use it as a natural basis for designing the trust management system itself.

It is time that not only a new and radical change is proposed in the we design our security, but also in the way we construct and work with access control. A more human an natural approach to this topic would mean a more natural way integration of such systems in the working environment. In this paper we propose such a system by creating a foundation in definitions and design based on a solid and understandable theory of humans as "living" control systems. Not only the integration of human-modelled control systems is a beneficial choice, the present literature on the field of trust management is often too narrow in its views and implementations to be of any use within the Jericho architecture. We need a system that can with reasonable certainty (100% is not achievable) give the answer to two of the important issues we have in granting access to data in this system: is this person who he says he is and is he trustworthy enough to be allowed access to that piece of data with this specific security level attached to it. In this paper we will design a system that is able to answer the above posed questions in way that comes as natural to its administrators and users alike.

[1] http://www.jerichoforum.org
[2] http://www.opengroup.org

Contents

List of Figures

Chapter 1

Introduction

As the Internet becomes an ever more integral part of our lives, it also has its effects on the way we do business. More and more we live in an open and fully connected world. The old way of thinking about securing computers and networks with perimeter firewalls and static access controls no longer suffices within this open, online world. The name of the project which is working towards a solution has a strong biblical analogy. The small city of Jericho bears a striking resemblance to our current fortress-approach to network security. Also comparable is the fact that when the outer wall of the city fell, it was conquered easily. This also holds true for the closed perimeter networks we still use today. Although for these networks, the situation is even worse, as security holes in the firewall, created for enabling online collaboration, make the risk even greater. It therefore does not require even a full perimeter breakdown for the protection of the security barrier to fail miserably.

1.1 Introduction to Jericho

The Jericho Forum [1], founded by the Open Group [2], describes the main goal of Jericho as follows:

> The huge explosion in business use of the Web protocols means that:
>
> - Today the traditional ”firewalled” approach to securing a network boundary is at best barrier flawed, and at worst ineffective. Examples include:
>   - Business demands that tunnel through perimeters or bypass them altogether
>   - IT products that cross the boundary, encapsulating their protocols within Web protocols
>   - Security exploits that use e-mail and Web to get through the perimeter.
> - To respond to future business needs, the break-down of the traditional distinctions between ”your” network and ”ours” is inevitable
> - Increasingly, information will flow between business organisations over shared and third-party networks, so that ultimately the only reliable security strategy is to protect the information itself, rather than the network and the rest of the IT infrastructure.

[1] http://www.jerichoforum.org
[2] http://www.opengroup.org

> This trend is what we call ”de-perimeterisation”. It has been developing for several years now. We believe it must be central to all IT security strategies today [25].

De-perimeterisation is not a goal by itself. If that were the case, simply de-activating all border-controls would suffice to achieve this. Admittedly, with all of the security holes already present, this would not make much of a difference for many networks. De-perimeterisation does however certainly not mean that all network-based security will be removed completely. Although the main focus will be on data, firewalls will still exist, as well as other security measures.

Before we can achieve the goal of de-perimeterisation however, a lot of other steps need to be taken to achieve adequate security to replace the perimeter firewall. Reaching this goal should therefore be seen as the final achievement in reaching a fully implemented Jericho architecture. In the next subsections we will look at the way Jericho security works. We will do this by showing the changes and differences between the Jericho method and current practise.

First, we will look at the change from securing the network and storage devices, to data protection. Secondly, we will show the difference between single-domain users and Jericho's user-centric model. To help explain the latter, we will also describe federated identities, as these are an important step between single-domain and user-centric models.

1.1.1 From Network-Security to Data-Protection

In this subsection, we will look closer at what the difference actually is between the outdated perimeterised and the de-perimeterised network architectures. The simple answer to this is just to switch off the firewall. However, any security conscious person would agree that removing all firewalls and other security barriers from the network would not be a good idea.

In general, it can be said that security barriers like firewalls will still remain on the network, but not on the perimeter of the network. Rather, securing individual systems will still be necessary. The Jericho forum therefore proposes to replace the perimeter-firewall with protection on a completely different level, data. The justification for this will be given below.

The Fortress-Approach

We have already drawn the analogy between current security practises and the city of Jericho. The fact that having a perimeter fence by itself is not the critical issue, but rather the fact that generally everything inside that fence is deemed to be trustworthy. The fortress-approach in security relies on this principle. The perimeter firewall is designed to keep things out and even to keep certain thing in, as well. Everything inside it is deemed secure and can be trusted. This creates some security issues by its own which are outside the scope of this paper. The fortress-approach is further known for its difficulty of accessing data and services by trusted parties travelling outside the main area of control. We will discuss this in section 1.1.2.

Securing Data

As stated above, it is very difficult in a perimeterised network to access data inside the perimeter from the outside. Normally users require either VPN-tunnels or other methods where security holes exists in the perimeter to allow access. To make this data accessible to anyone who is authorised to, yet at the same time keeping it secured in storage, during usage and during transit, we need to secure the data itself.

This is done by first classifying the data in different levels of access. For example, a simple white paper does not need any encryption or access control at all as anyone is allowed to view it anyway. But highly important strategic management or financial data does. Examples of this can be found in the media almost daily. Lost laptops and USB-sticks with credit-card data, medical information of patients, security arrangements for military operations, the list goes on and on. These examples alone should be explanatory enough to show why data protection is absolutely necessary and why simply securing the network is no longer sufficient.

How data is classified is outside the scope of this paper, however the resulting access levels are of interest as they will play a major role in deciding what, and more importantly, how much information is required from the user to be granted access. In general, it should hold that the higher levels require more authentication information and a higher reputation.

1.1.2 From Single-Domain User to User-Centric Authentication

Now that we have defined why we need protection at the data-level instead of perimeter firewalls, we need to look at the other part of the puzzle, user authentication. We will specifically look at from where and for what the user is authenticated. For this reason, consider a standard authorised user on a corporate network. Additionally, that the user is also outsourced to a client of this company.

Single-Domain Model

In this model the user, defined in 1.1.2, is allowed to log on to the corporate network from his own company. For this, he probably will use a username and password to authenticate to the network or different services. Problems arise when the user is outsourced to the client however. The username and password he has are no longer valid anymore. They are still valid of course on the company network, but not at the network of the client company.

To be to granted access at the client-location, a new set of credentials needs to be created. So our user is granted a new set of username and password. This is still manageable with only two sets of credentials, but what if this user is working at four or five companies. It is easy to see that the credential management for the user is quickly getting out of hand.

Federated Model

There is a big problem when users tend to switch from network to network. On all networks they need a new set of credentials to be granted access. This is difficult to manage for hundreds or even thousands of users. For these reasons, companies could setup an identity federation. This would mean for our example user that he can easily switch to the other network and still be able to authenticate using the same username and password he was issued by the company he works for. However, if the user goes to a company with which no federation agreement exists, we are back to square one, being the single-domain situation.

User-centric Model

What we actually want is that our user is able to access the network and systems he is granted access to from where-ever he is getting onto the Internet. For this approach, the company would be required to sign federation agreements with any possible Internet service provider or

company his employee could ever try to request access from. Apart from being a managerial nightmare it is simply impossible to maintain.

We need something more generic, yet secure enough to allow this form of access management. The Jericho Forum solves this problem by using a user-centric approach. For our user, this would mean that he can request access by a local authentication agency. This agency will then try to establish if the user indeed is who he claims to be.

For this purpose, it may contact a similar authentication agent at the home company of this user, present it with the information the user supplied and ask for verification. The difference between federation and user-centric lies in the fact that no previous contracts were signed between the local agent and that of the company the user is working for. There are several more issues related to these models than discussed in these examples. However, only the user-centric model is of interest to us in this paper.

1.2 Jericho Commandments

The Jericho commandments define the principles of the Jericho system, as well as, the areas it consists of. In effect, these are the general rules for building a Jericho compliant security architecture [26].

Here as well, some biblical analogy is unavoidable, although as in this case there are 11 commandments whereas the Bible only counts 10.

The commandments are divided in the following groups or areas:

- Fundamentals, commandments 1 to 3
- Surviving in a Hostile World, commandments 4 to 7
- Identity, Management and Federation, commandment 8
- Access to data, commandments 9 to 11

A full listing of the Jericho Forum commandments can be found in appendix D.

From these commandments, commandment 8 [1] is the most important one to keep track of. Although commandments 6 [2] and 7 [3] play a major role in the requirements of the trust management system.

The position paper of the Jericho Forum [26] concludes with the following:

> De-perimeterisation has happened, is happening and is inevitable; central protection is decreasing in effectiveness.

The concept of ”de-perimeterisation” is not something new, it is happening now, the fact that there is no fully working solution for it yet, makes this research all the more valuable and necessary.

With this paper we hope to contribute to such a system and create an integral part to a fully usable Jericho-readiness road map.

[1] Commandment 8: Authentication, authorisation and accountability must inter operate / exchange outside of your locus / area of control

[2] Commandment 6: All people, processes, technology must have declared and transparent levels of trust for any transaction to take place

[3] Commandment 7: Mutual trust assurance levels must be determinable

1.3 Capgemini Jericho Research

In January 2007 a group of students at Capgemini started work on defining a new way of thinking about security based on the Jericho architecture which is explained in the next section [1].

This group laid the foundation for further studies. In this thesis we will build upon this previous work and will extend it by defining a trust management system. The work done by M.A. Bruning in [33, 34] is the basis of the trust-broker design which is a good starting point. The trust management system we will design in this paper, will be an important part of inner workings of the Jericho trust broker.

The trust broker is dependent on a good implementation of end-point security and authorisation described here [57], Authentication and accounting described here [1] and secure communication and encryption described here [56]. Although these dependencies are crucial, it is outside the scope of this paper and will only be referenced where needed.

1.4 Why Trust Management?

The obvious and simple answer to this question would be: because we have a trust broker. However, we could then have sufficed with a simple but not trust related solution like a PGP- or X.509-style certification system.

The challenge we have set ourselves in this research is to find a way to make a real trust-based access-control system. Based on the human notion of trust and build such that it not only will take a decision of access is allowed or not, but possible use various degrees of access and even try to resolve issues arising in the evidence available either by setting a lower level of access allowed and/or using alternative methods to acquire the missing information by other means.

Trust is something important in a lot of daily issues, even very simple ones where it is something very implicit. It is our goal to see if we can define an access-control system that uses this aspect and therefore rises above the commonly available solutions that are on the whole nothing more than your old style access control list (ACL) [2] [3].

In the end we believe that only such a system will be able to be scalable and reasonably secure enough to stand up to scrutiny in the future of a fully open and transparent global Internet.

1.5 Research Questions

In this thesis we will answer the following main question: How can we define a trust management system within the Jericho architecture?

As the scope of this question is quite broad, the following sub-questions are defined:

- How can we use Perceptual Control Theory (PCT) to define digital trust?

[1] `http://www.onewalldown.com`

[2] ACL: a static file with access decisions per line dependent on hostname, IP-address or range or username

[3] Even other systems in essence are ACL's, although they are mostly dynamic in nature or use other means (like certificates) to get to the access decision, because the outcome mostly is a binary choice (allowed or denied), the basics (although more advanced) behind the systems are not much different than that of an ACL

- What are the requirements for defining a trust management system?
- How can we define a trust management system by using its requirements and our digital trust definition?
- What are the security requirements, issues and pitfalls in the trust management system in the context of the Jericho security model?

1.6 Research Methods

For the completion of this paper a certain set of research methods have been used. Most prominent of these is the literature read and referenced in the bibliography. Complementary to this has been a set of interviews with leading figures in the field of PCT. Noteworthy names here are William T. Powers the founder of PCT, Dag C. Forssell and Frans X. Plooij.

Also brainstorm sessions on direction, vision and scope of the research have been held with Alina Stan. Further information has been obtained in discussions with Adriaan Bruning who did the preliminary work on Jericho Trust Broker which can be found in [33, 34].

The initial scope, research direction and pointers to PCT as well as two articles on the implementation of PCT have been provided by Marco Plas [37, 38].

1.7 Thesis Outline

In this chapter we have given an introduction to Jericho and the user-centric model. We have also introduced the Jericho Forum commandments focussing on the important ones for this paper. We concluded this chapter with a reference-section on the research done by the Jericho-team at Capgemini.

Chapter two will introduce PCT and use this theory to give a definition of trust. This definition is vitally important for the system we will define in chapter five.

Chapter three gives an overview of past research in trust management. It will also list the useful points from this research which will be used in chapters four and five, but also the shortcomings which we need to define.

Chapter four defines the requirements for a trust management system. It will use the starting points defined in chapter three as well as the shortcomings.

Chapter five finally will combine the definition of trust given in chapter two and the requirements defined in chapter four to define the trust management system itself.

Chapter six finally will draw conclusions and list possible further points of interest. We will formulate possible research area's that were outside the scope of this paper, but are important to further the implementation and adaptation towards a usable Jericho Trust Management System. The chapter ends with the used research methods and acknowledgements.

Chapter 2

Perceptual Control Theory

Behaviour Controls Perception

In this chapter, we will look at Perceptual Control Theory (PCT for short) which at first might seem to have nothing to do with the main topic of this paper. However, one subject on this paper has everything to do with theories from the social sciences, namely, trust. Why we have chosen PCT is outlined nicely in [19]. Trust is something which is hard to describe in a pure digital way. But because we are designing a trust management system, we do need such a definition. PCT, because of its roots in engineering, can be used to bridge the gap between our digital design and the human notion of trust.

In this chapter we will first describe what PCT is in general, followed by an example. We then will describe the hierarchical nature of PCT. We end with a descriptions of trust, reputation and risk which are used to define trust in the context of Hierarchical Perceptual Control Theory (HPCT).

2.1 Introduction to Perceptual Control Theory

PCT is a theory used in the social sciences to describe human behaviour. However, it has its roots in engineering. PCT has been invented and possibly re-invented from as early as 240 BC up to the 1930s, when the general principles were finally worked out. In its current form it is inspired by the work of the early cyberneticists (Wiener and Ashby, primarily) in the 1950s. Until now we referred to this theory as PCT, the full implementation and creation of Perceptual Control Theory is actually the work of one man, William T. (Bill) Powers [41]. It has minimal recognition in the social sciences which seems is more because the theory is quite young. Dag C. Forssell has written two good articles explaining the theory [20, 21].

In the remainder of this section we will describe the basics behind PCT. In the next section we will explain the hierarchical concept of PCT. In section 2.3 we will illustrate the workings of PCT with a practical example.

2.1.1 The Meaning of Control in PCT

Whilst the discussion of the mere meaning of a word is far outside the scope of this paper, we do have to define what we actually mean with the word ”control”. Normally, control is associated with many things that aren’t necessarily the right interpretation. For instance, the controls of

an aeroplane actually do not control the plane itself, a pilot does this by operating the controls. The same holds true for a control experiment which does not control anything, it simply is a reference for the experimental data. Going back to the aviation industry, an air traffic controller does seem to control planes, although if you look closer even this is not the case. He or she simply requests the pilots to act in certain ways.

While the last example is closer to the meaning we will give the word ”control”, a formal definition is required:

> A is said to control B if, for every disturbing influence acting on B, A generates an action that tends strongly to counteract the effect of the disturbing influence on B.

The definition is taken from [41]. This is not a complete definition, but a useful part and it will suffice.

If we look back at the examples given above, the pilot controlling his aeroplane is a true example that fits the definition. We can illustrate this by look at a landing passenger jet on the runway with a gusting cross wind. The pilot needs to counteract this disturbance of his flight path to land safely. Besides responding to the disturbance, the other part is the fact that he or she is actually controlling the direction and speed of the jet. If we would have left this by responding to the disturbance, we would have told only half of the story.

A larger example of PCT is given in the in section 2.3.

2.1.2 The Control Loop

Now that we have the basics, let us put everything together and explain in general how it works. Note that section 2.3 will further illustrate PCT by applying it.

We now define two different forms of perception, namely, a reference perception and a present perception. The reference perception corresponds with your mental image, specification or purpose of what you actually want to accomplish. The present perception, on the other hand, is the mental image of what is actually occurring.

The control system works by continually comparing these two perceptions. The difference between these two perceptions is known as dissatisfaction. This ultimately will lead to action which will cease when the present perception correspond with the reference perception.

> Text excerpt from [41] by figure 2.1.
>
> The grey bar across the diagram separates the active control system (above) from the environment (below). The red lines show the closed causal loop of control. Small circles in the environment show where physical variables can be measured: an output quantity, an input quantity, and a disturbing quantity. The green lines indicate effects of independent variables: the reference signal and the disturbing quantity. The Input Function converts a sensed variable in the environment to a signal representing it inside the system; the Output Function converts the error signal into a physical effect on the environment.

What we have defined here is actually a ”living” control system (i.e. human purposeful behaviour). This system will shape the world around it in a way that for every outside influence (disturbance), it affects something it its environment it has a reference perception for, in such a way that the present perception matches the reference perception again [13]. The net result of this is a closed causal loop defined by interacting elements and signals.

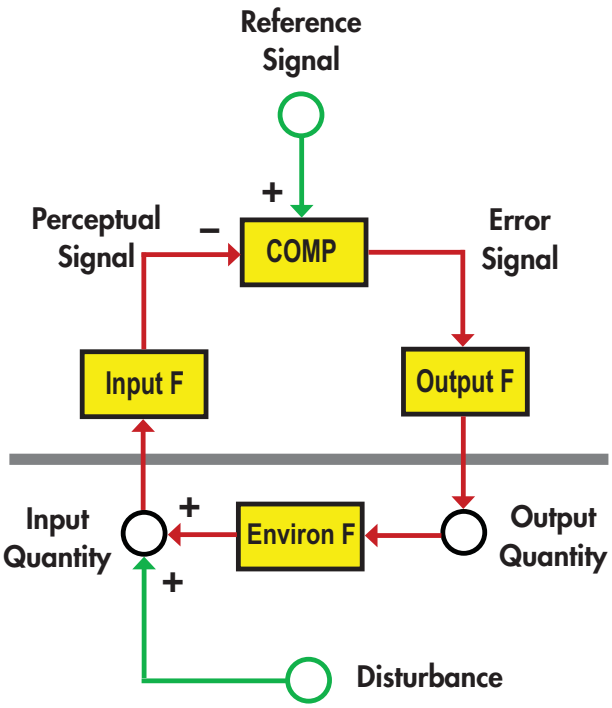

Figure 2.1: The Basic PCT control system [41]

We have already defined the three signals, however with slightly different labels. This has been done often for different purposes ranging from a more plain explanation or mathematical reasons. We are going to look at it from the engineering standpoint by defining signals and functions.

First we redefine (or rephrase) the three signals. The reference signal, reference perception, (r) specifies the value to which the perceptual signal, present perception, (p) must be brought [22]. The reference signal could be described by words like want, need, hope, desire, wish etc. This signal is set from the inside and is independent from any sensory input whatsoever. Low-level reference signals in a ”living” control system (i.e. a human body) will have build-in intrinsic values for all kind of body functions [22]. High-level reference signals include memory as you recall certain experiences and want to experience them again. It is at these high conceptual levels that things like honesty, friendship, science and love are constructed. We get back to these concepts in the next paragraph and to the hierarchical aspect in section 2.3.3.

The perceptual signal meanwhile is comprised of sensory input. This signal will ultimately be equal to the reference signal but only if control was indeed successful. For sure not all attempts at control are [22]. An example of this is when you push against a closed door. The reference signal will indicate that it needs that door to open. However, because it is locked, the perceptual signal will not reach this state after the attempt is made.

The third and final signal we need to define is the error signal, dissatisfaction, (e) which is the result of the comparison function (c) which has this signal as its output signal. The comparison function has the other two signals, perceptual- and reference signal, as its input signals. The difference between these two signals is the error signal. By engineering convention the reference signal is marked with a plus-sign whilst the perceptual signal is marked with a minus-sign [22].

We have already associated the error signal with the term dissatisfaction. Other terms that would fit here are unhappiness, unease, something is wrong, hunger, thirst or fatigue [22]. Any emotional state may fit as well as we will see in the next paragraph. One noticeable effect in ourselves as ”living” control systems is that the error signal is more perceptible than the reference signal. This can be easily illustrated by i.e. the error signal for hunger which is clearly noticeable and indicates a low level of blood sugar. The accompanying reference signal indicates the required level that your body wants to have. However, you do not know what this level is. The only thing you know is that you need to eat to get rid of the hunger feeling. You do not even realise that it is the low level of blood sugar that is causing it. Or in other words: the perceptual signal which holds the current value of blood sugar does not correspond with the reference signal holding the required level.

The last remaining element to define is the output function (f). This function has the error signal (e) as its input and will either output signals to muscle fibres or glens in your body or as reference signal to a lower-level control-loop when it is higher up the hierarchy [22]. The output of (f) is described by the action/output quantity (qo) and commonly referred to as behaviour.

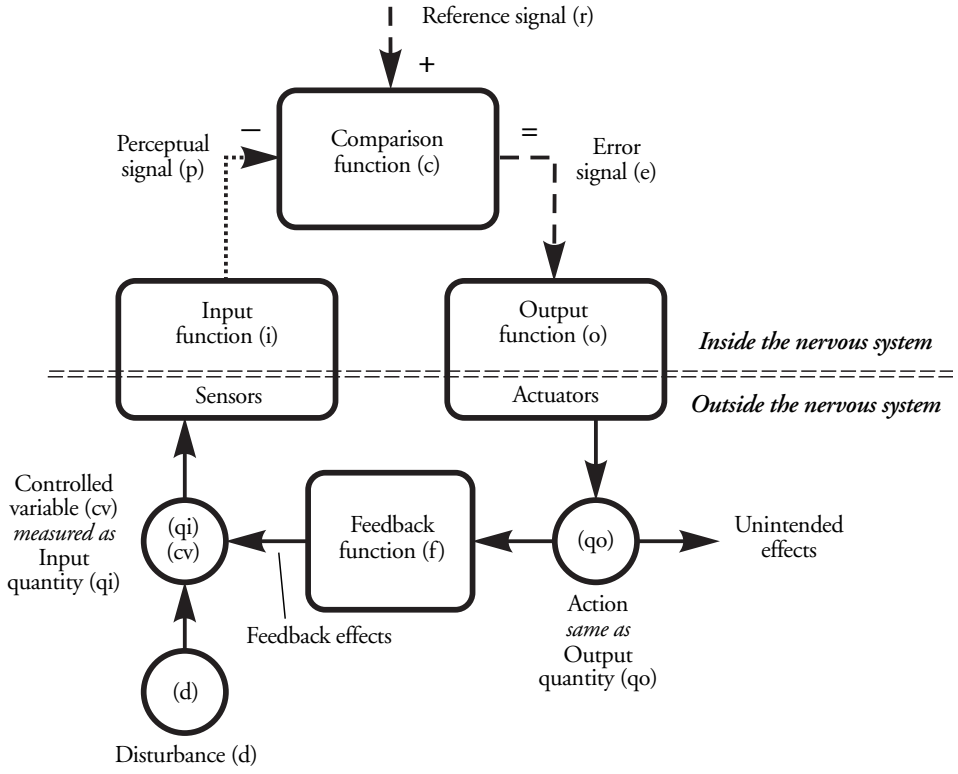

Figure 2.2: PCT: Once Around the Loop [22]

Text excerpt from [22] by figure 2.2.

Note: This illustration can be seen as a single elementary control system, consisting of a few neurons and muscle fibres acting at the interface with the environment, or as a summary of an entire hierarchy, thousands of control systems at many levels, acting in complex ways on the environment. Arrows in the nervous system indicate neural signals carrying information from one function (neural network) to another. Arrows in the environment indicate physical links that give the output of

one function a physical influence on a physical variable. The circles show where physical variables are, or where they could be measured. Functions in the environment usually indicate physical laws that determine how physical variables at the output of the function depend on physical variables at its input.

The last pieces of the puzzle are the input quantity (qi) which is also called the controlled variable (cv) and the feedback function (f). The controlled variable is what it all is about. The output quantity may have other and unwanted site-effects, but its effect on the controlled variable is what we are interested in and what ultimately must match the reference signal. Apart from our effect on the controlled variable there may also be outside disturbances (d) that have effect on its value i.e. a crosswind on the path of your car whilst driving (see section 2.3.3). The feedback function (f) describes the effect your action has on the controlled variable.

Any result of a control loop like this might be effective i.e. when you eat something the feeling of hunger ceases, ineffective i.e. when trying to push against a closed door or indirect i.e. when complimenting a colleague hoping for a favour next time round. Certainly not all control attempts are successful [22]. From this description you may get the idea that the control loop we have described is operating in a step-by-step motion. This, however, is incorrect. All signals and functions operate in a continuous flow where everything influences everything else all the time.

2.1.3 How To Experience Control

An interesting experiment is described in [40] which demonstrates what control is and why only looking at behaviour, we come back to this in section 2.3, yields the wrong conclusions.

This experiment was included in Powers "Behavior: The Control of Perception [39]. Philip Runkel adapted it in his book "Casting Nets and Testing Specimens" [50]. The illustrations used here are taken from another article by Dag Forssell that closely follows the text of Runkel's book [18].

The experiment, involving two rubber bands tied together with a knot, demonstrates the workings of control. The meaning of the word "control" is exactly as we defined it in section 2.1.1. In this section we describe and illustrate how this can be done.

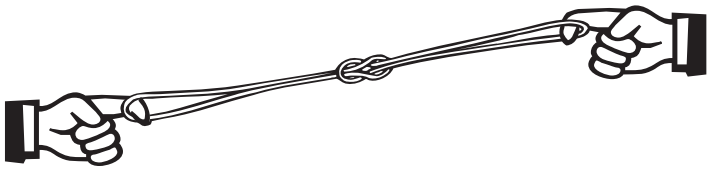

Figure 2.3: Rubber band game, starting position [18]

In this game, any disturbance on the environment has a direct and obvious influence on the controlled variable. The only thing that is required are two rubber bands joined together with a knot and two players. The starting situation of this game is defined by both players hooking one finger into the end of one of the two rubber bands. One player is the experimenter. His or her job is simply to move their site of the joint rubber bands around in any direction they may chose to. Whilst your task as player is to chose a simple reference point, marker, and counteract every move the experimenter makes to keep the knot over the chose marker. If the experimenter moves too fast or to abruptly, simply ask her to slow down.

The experimenter's task has one additional element to it, which is the actual learning part of this game and that is to notify the opponent, you, when experimenter can explain what is causing

you to do what you do. You have reached the objective of this game if the experimenter indeed does notice that you are not simply countering every move that is made, but are actually trying to keep the knot over a chosen marker. This again demonstrating that simply analysing the behaviour, your movements, does not explain why you are moving this way, PCT does.

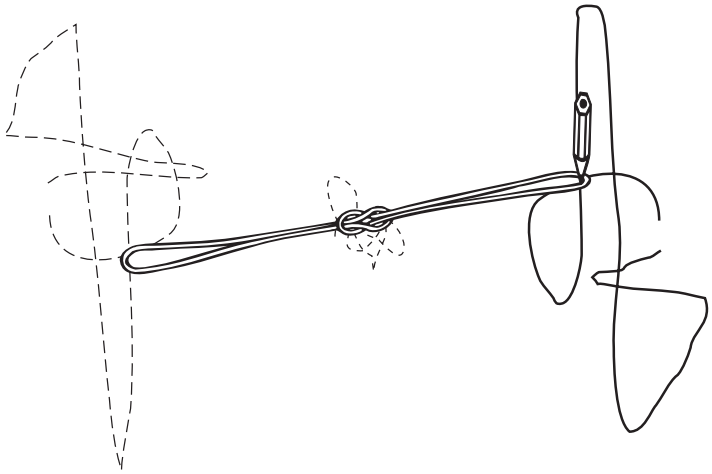

Figure 2.4: Rubber band game, tracing action [18]

There is no winner to this game, although if you already understand PCT and you succeed in making your friend discover the workings of control as well, you may grant yourself a moral victory. Obviously not everyone will notice this directly or even notice it at all. Explanations given may include that you are simply mirroring their action or even imitating it or other words to that effect. Some will even add this more forceful and say you are acting in opposition. Almost all will imply that they are the cause of your behaviour. Some indeed may notice that you are keeping the knot stationary above a certain target, which is an excellent observation. It however still is no explanation. However, in the end most will still not be able to explain it and ask you to do so. At which time you explain to them what your goal was, that is keeping the knot as close as possible to the chosen marker.

Actually what you have been doing is simply moving to the opposite direction of where the knot was going. The fact is that whatever your friend was trying to do is besides the point. You did not even care about it. Even so, if you would not have been able to see your friend at all, your actions would have been exactly the same. Replies to this explanation may vary as well. Some will simply still not believe you and accuse you of being devious, most will certainly be surprised that they missed something that obvious and others will find some of their previous believes so shaken up that they will keep thinking about it for days.

Now let us put this game in to perspective and link it to the terms of PCT. Your choice of the visual marker in this game is the internal goal you have set (reference signal), the location of the knot itself is what you perceive (perceptual signal), the knot is the controlled variable, any action you take to keep the knot above the marker is generated by the error signal that is automatically created in the comparison between the reference signal and the perceptual signal and finally the action of the experimenter is the disturbance on the controlled variable in this simple environment.

So what we have here are all the required elements of a closed causal loop as defined in PCT. As a side effect, notice that we implicitly also explain the title of this chapter. Your behaviour as generated by the disturbance of the experimenter on the location of the knot relative to the marker and your goal to keep it above that marker, is directly influencing your perception of the location of the knot.

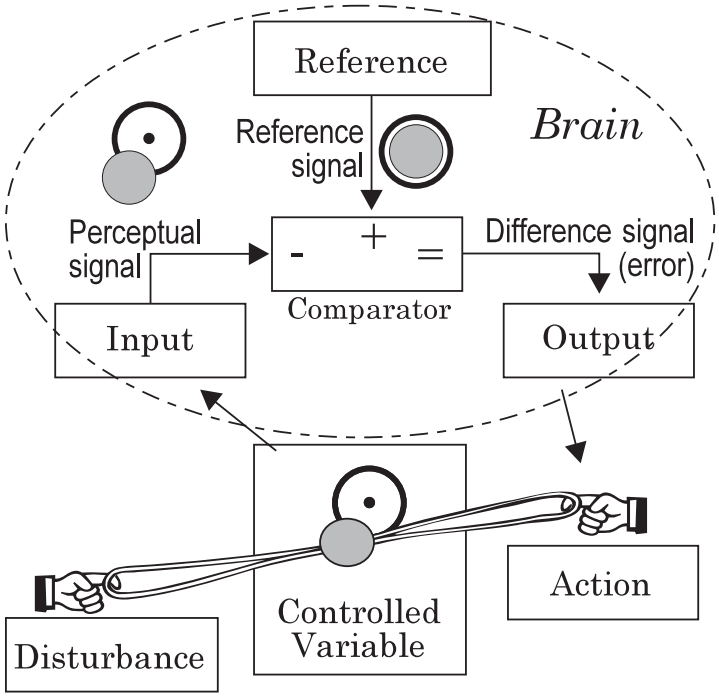

Figure 2.5: Rubber band diagram [18]

Two additional experiments are possible that demonstrate conflict or cooperation. These are also outlined in [18], but will not be discussed here. However, the last thing you may do to satisfy the curiosity of your friends that found out what control actually means is, to reverse roles and let them demonstrate they really do understand.

2.1.4 Explaining Emotions with Perceptual Control Theory

Emotions are confusing experiences, specially because they seem to be cause and effect at the same time. The age old question is if the emotion is there before or does it arise as result of a certain event i.e. do we know there is danger because we feel afraid and flee or do we perceive danger and feel and feel afraid as result of this. For the model of emotion in PCT we use close causal-loops as the rule, not the exception. Sequential causality is not enough as we need to have a clear concept on how control works and be able to use local causes and effects existing in the system at various places at the same time [43].

In a model using closed-loops we recognise that an experienced emotion is a set of inputs, perceptions we would call ”feelings”, and a set of output-cased changes in physiological effects. Beyond these basics there are also reasons based on what we seek and what we avoid. The emotions associated with an experience we seek are given ”good” names, ”bad” names are preserved for the emotions associated with experiences we avoid. However, if we only look at the accompanying sensations, they are similar [43].

The cause of an emotion is a reference signal somewhere in a high system-level which sets a low- or high-level of some perception. If this matches the current perception there is no emotion as there is no error to correct. However if they do not match there is an emotion. Emotions are therefore related to error signals in high-level control system. This error signal is then converted into new reference signals for lower-level control systems [43]. Somewhere in the mid brain these signals cascade where one part goes into the motor system that produces

action, the other branch passes into the physiological control systems, the life-support system of the body. This second branch alters the state of the physiological systems according to the action produced by the first branch, the behavioural branch. The second branch is the branch in which feelings arise, apart form the feeling of muscle activities.

Under normal circumstances, the error signal that gives rise to actions and feelings is either caused by a change in reference signal or when a disturbance which chances some perception. When an error results in a change of action, this is simultaneously supported by changes in the physiological system. As these changes happen more or less at the same time, both systems are in balance [43].

In PCT emotion is seen as part of the central nervous system and the physiological systems it uses to achieve its goals. The behavioural part of the hierarchy is constructed of many different levels of which only some are in contact with awareness. The others will simply keep on function as they did on the last time they were reorganised. A change in behaviour can result form any error signal without any regards to consciousness at all. Therefore an emotion can occur at any time there is an error signal whether we are acting conscious or unconscious to reach a certain goal or correct an error at any level [43]. However, this only counts for significantly large error signals, it seems there is some kind of level above which these errors are described as emotions.

Emotional thinking or emotional behaviour are simply normal behaviour but for a topic or situation that is important to the person. Even so important that strong action is required and even small error signals may not be tolerated. That this kind of the behaviour that is suggested by the continuation of the emotional state. However, it is unjustified to dismiss ”emotional” arguments for that reason. It may even be the unemotional argument that is defective as it does not contain any errors whatsoever [43].

2.2 Hierarchical Perceptual Control Theory

It may not come as a total surprise that the complete model of PCT does not consist of one single loop, but in fact is a complete hierarchy of linked but separate control systems. Or, as in the case of a living control system like ourselves, even multiple once all working in parallel. In the previous section we have, either explicitly or implicitly, already hinted at the hierarchical concept within PCT without explaining what it is. In this section we explain what it is. We will also shortly introduce the concept of the ”method of levels” which is closely related to HPCT. In section 2.1.2 we refer to the hierarchy of PCT a couple of times and figure 2.2 can be see as ”displaying an entire hierarchy of control loops”. One clear example of hierarchical nature can be given by explaining its various levels of perception.

> Text excerpt from [17] by figure 2.6.
>
> The vertical dimension is ”Levels of perception”. Starting from the bottom: a low-level input, a neural current created by a nerve ending, ”tickled” by some physical phenomenon in the world, such as light falling on a single cell in the retina, is combined with other inputs, creating a perception signal at a higher level, which is in turn combined to create a signal at a still higher level. At the higher levels, a branch of the perceptual signal can be recorded in memory and later played back as a reference signal.
>
> The horizontal dimension is ”Examples of perception.” At the lowest levels, we perceive light, vibration, pressure, temperature, joint angles, tendon stretch, smell, taste and physiology (which we sense as a part of feelings). The highest percep-

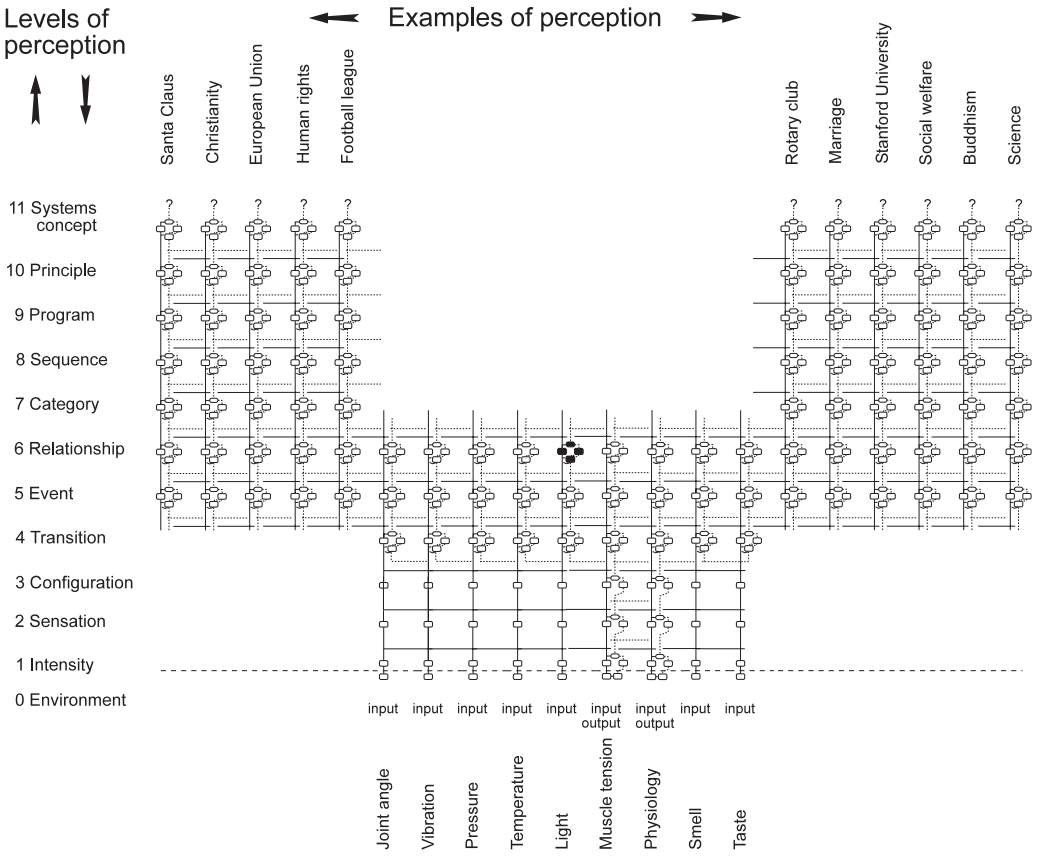

Figure 2.6: PCT: Levels of perception [17]

tual levels are called systems concepts. These are descriptions, explanations and models of the world, in many areas of knowledge, which we learn and decide to believe in.

In section 2.1.4 we explained the model of emotion in PCT. In his book ”The Method Of Levels” [8] Tim Carey talks about the application of PCT in psychotherapy. He explains that the method of levels (MOL) is not a new way of getting patients better, its only means used to help his patients getting themselves better. This is in line with what we have seen in section 2.1.4 in which we stated that trying to change somebody his behaviour only leads to massive error signals and therefore will not work [43]. The method is based on the attempt to do the only thing necessary and nothing that is unnecessary. The idea behind this is that people will get themselves better by reorganising their internal control systems.

Reorganisation and awareness seem to be linked somewhere. Awareness seems also to be something fluid, but the hole concept of awareness is something that is not that well understood but undoubtedly does exists [9]. Because of its fluid nature, it is probably harder to make awareness stay still than to get it moving. Awareness, like reorganisation, is something private happening in a persons own individual neural hierarchies. Changes in the awareness of a person can be noticed only when i.e. as a disruption during a conversation when they break their on dialogue. If that happen they should be asked about it to see if their awareness shifted up a level in the hierarchy, if so the hole process is start all over again [9].

When conflict arises, the solution is found in ”going up one level”. This is not only the main principle behind MOL [8], but we have seen the same effect in the emotion model in PCT [43]

where a higher level control system needs to resolve a conflict by changing its output signal (which is the reference signal on the level the conflict is situated). The input for this higher-level control system is the combined perception of all lower-level control systems, thus not only the one directly beneath it. This principle of ”going up one level” is therefore seen in emotion resolution as well as in the method of levels. Which is not that surprising as it deals with emotions.

We briefly mentioned reorganisation. Let us now explain what it is. Reorganisation takes place when an person is not controlling is world well enough which leads to a massive difference between what the person wants and what it is perceiving. This is defined in PCT as a large error signal or dissatisfaction. These, often chronic, error signals are undesirable. They are perceived by a very basic 'dumb' biochemical control system which makes random changes in the organisation of the control hierarchy [17]. Reorganisation takes place on basic neurological and biochemical level as well as on the higher conceptual and systems levels. Reorganisations may even be sudden and chronic errors may not even be resolved by only a single reorganisation, multiple states may pass before the reorganised hierarchy is able to cancel out the chronic error or the person dies [17]. Mild reorganisations may be related to a person solving a difficult problem.

The last aspect we need to discuss is the timing aspect between levels. Obviously it is impossible for every level to work at the same speed. As higher levels in the hierarchy use the error signals and combined perceptions of lower levels as input signal and send output signals, actions/behaviour, back down the hierarchy. In their book ”Introduction to Modern Psychology” [45] Robertson and Powers describe four of the lowest hierarchical levels in a human ”living” control system. They use these to illustrate the timing aspects of multiple levels in a control hierarchy. This aspect is very important for the final design, but will not be further discussed in this paper. However, more can be read in [16] and appendix C.

2.3 Perceptual Control Theory: an Example

In this section we will give an example of how to apply PCT. We also will compare it to two theories, which are mainstream within the social science, to illustrate why PCT is a realistic choice for defining digital trust. Please note that we are not trying to give any formal proof of PCT, we only illustrate the working of PCT. Anything else is outside the scope of this paper.

2.3.1 Situation Sketch

The example given in this subsection bears a close resemblance to that given in [41]. Even ”that pesky crosswind” is still a factor.

Consider the main entrance to the Capgemini building at the Papendorpseweg 100. There is a big square in front of the main entrance and the visitor parking. It slightly slopes down towards the building. The result of the latter, is that if you would miss the entrance, you would wreck your car. Now consider yourself driving along the road and wanting to turn into the main square described above. There might, or the might not be, a crosswind blowing. If it is, you would need to steer against it to avoid hitting the side posts of the entrance-gap.

In the following sections we will observe a driver carrying out the action of driving into the main square. We will look at this situation twice, once with and once without the crosswind.

2.3.2 Mainstream Theories

In this section we will look at two mainstream theories that are mostly used in the social sciences. These are:

- Reaction to external events and circumstances
- Cognitive planning or computing

We will illustrate for both of these theories why they do not work, or only offer a partial explanation.

We will only compare these theories with Perceptual Control Theory. As can be seen in the following illustration, Both theories describe a different aspect of human behaviour. As we will show, this is precisely the reason why both theories are insufficient for our purpose.

Reaction to External Events and Circumstances

This theory concerns itself with external stimuli. As in our case, the Capgemini signboard triggers the response for our driver to turn into the main square in front of the building which leads to reinforcing consequences. So, every time the driver turns into the main square, the likelihood increases for the driver to make the same turn the next time when he sees the Capgemini sign again. However, as there are other stimuli present as well, the driver might not turn into the main square every time. But when he does, this is the explanation for it.

The trouble with this explanation is that it only explains why the driver turns into the main square when he sees the Capgemini signboard. It does not explain how the he comes to this action or what action needs to be taken. Our example in section 2.3.1 explicitly lists the possible presence of a crosswind. Because on alternate days this crosswind might be blowing and if so, the direction may vary, as well. The result of this is that the action our driver needs to take, the pressure required on the steering wheel, varies in the changing wind conditions.

Somehow the same stimulus, being the Capgemini signboard, needs to trigger a different response in accordance with the wind condition at that particular moment. It is not a question of simply choosing the right sort of response, it has to be correct to get the car around the corner safely. So we do have somewhat of an explanation why the driver wants to turn the car into the main square, but how he does so and how he responds to different wind conditions, is not explained at all. There are possibly more issues with this theory, but this will suffice for the scope of this chapter.

Cognitive Planning or Computing

Contrary to the previous explanation, this theory is more concerned with what happens, or more precisely, what needs to happen, to turn the car around the corner. Why the driver actually wants to do so, is completely left unexplained. The practical issue with this theory is, that the driver is expected to have knowledge of at least the speed of his car, the condition of the road and his tyres, the car's mass and inertia and any other factors such as muscle tension and the properties of the steering linkage of the car itself. All this information combined, the driver now needs to calculate what muscle tension to use to achieve the desired result, being the car turning the corner.

Even this would give a mathematician, without a blackboard, a headache, but we are not done yet. Above all these differential equations to be solved, there also is the matter of the

crosswind, whose direction and speed need to be taken into account as well. Apart from making the equations even more difficult, the driver inside the closed car is also not able to sense the crosswind, let alone calculate speed and direction. Debris flying across the road may give some indication, but it will not provide enough information to accurately calculate its influence on the movement of the car.

Our brain is a wonderful and complex analog computer system, however, it is not a high-precision calculation device. Also, its sensors fail the precision necessary to make these kinds of calculations. So, to assume that the brain could generate the correct signals and signal patterns to turn the car into the main square of the Capgemini building is absolutely unrealistic.

2.3.3 Perceptual Control Theory

While the combination of the two theories of the previous section may seem to give a complete picture, in this section we will see that it can be explained more effectively and realistically, without requiring massive calculations by our driver's brain. We will show that both theories are contained within the definition of PCT and explain why each is inadequate. PCT is not a third alternative, one of three equals, but it is a larger more all-encompassing functional explanation.

In section 2.1, we have given definitions for the meaning of control and the control loop. In this section, we will use both definitions to illustrate why and how our driver makes all necessary decisions to get his car successfully into the main square of the Capgemini building.

First of all, let us look at the last stretch of road the driver takes before he turns into the entrance. We can define the perceptual signal of the control loop as the view the driver has through the main windscreen, the reference signal can be defined as the notion the driver has of a stretch of road and the line he should take on it. Now, if we have a crosswind, the car might be pushed off course. In this case, the view in the windscreen changes.

When this happens, the reference signal and the perceptual signal will differ creating an error signal. The driver needs to turn the wheel in the opposite direction of the cross wind to get his car back in the right direction. With defining this last part, we have completed the control loop for this instance. In a similar fashion, the turn into the main entrance can be defined. Of course, a complete description would include a long list of similar control loops deciding on muscle tension to turn the wheel or the head of the driver and such more. But for simplicity's sake, we will leave it as is. Please note that it doesn't matter if that pesky cross wind is blowing or not. If it is, the error signal will be greater, so the resulting action to cancel it out will also be greater.

Now that we have defined how the driver is able to turn in to the entrance, we should look at why he does so. To be honest, we do not know why the driver does this. Whatever the reason is however, it is again a result of dissatisfaction or disturbance signal. Maybe the driver's girlfriend works at the company and he is driving over to pick her up. In which case, her absence is the error signal and the action to pick her up from work is the required action.

In general, it can be said that for every decision a person takes, a similar reasoning can be done to explain how it works. These can be millisecond decisions (e.g. to blink an eye) or long term ones (e.g. to get a college degree). In all cases, it is possible to distinguish the reference, perceptual and error signals and the resulting action that should be taken to counteract the error signal again, which should bring the reference and perceptual signal in sync again.

A nice example why just observing an outcome and then drawing conclusions how it was reached is simply not good enough, is given in [40].

2.3.4 Explaining Behaviour

In section 2.1.2 we already stated that the output of output function (f) is named as the output quantity (qo) which is often referred to as behaviour. Even so, in the explanation given in this section we have been talking about nothing but behaviour as well. And finally the title of this chapter refers to it as well.

It is about time we shortly explain behaviour in the context of PCT and as specially in reference to the title of this chapter which states that ”behaviour controls perception”. All behaviour by a human being, also called actions, are observable by the outside world. This leads to the fact that only that part of the living control systems is paid any attention whilst the rest of it is largely misunderstood [22]. The problem with this approach is that people try to change the behaviour of others, which in almost all attempts will lead to massive conflicts (and error signal creation) in the target person. In the end the desired result is not reached at all. The idea however that behaviour is controlled by the individual and can be modified by others is widely accepted [22].

The question may arise if people are even aware of their own behaviour in such a way that it can be argued that they really can and do control it. In PCT action, and therefore behaviour, follows from the result of the comparison function (c) which compares the reference signal (r) to the perceptual signal (p) (see figure 2.1). Because of this, behaviour follows out of this comparison automatically. Therefore the answer to the above posed question must be no. The behaviour that is automatically generated has effect, amongst outside influences, on the controlled variable (cv) which is what is perceived by the input function (i). If you look at this closely it is now very simple to see that ”behaviour controls perception”.

2.4 Defining Trust in Terms of Perceptual Control Theory

What makes us trust each other and what not? We could talk about this for ages as trust is not something clearly definable in a set of mathematical equations. However, with PCT we will show that it is visible to at least come to some form of definition. The step thereafter, switching from the social sciences definition to the digital, is made in 4.2. It might look quite easy to come up with a nice control loop and some trust value to control and call it a definition of trust in the sense of PCT, however that will not suffice. Although in the end it may still come down to something quite similar.

In the next section we will look at some relevant definitions of trust. This is followed by a definition of reputation, in our opinion the only measurable element of trust. We then will discus risk as the flip side of a coin from trust. We conclude by defining trust with the help of PCT.

2.4.1 What is Trust

”Trust is an elusive notion” according to [28] and if you look at the definitions that are available they are either vague or just descriptive.

By looking a dictionary definitions [1] of trust, it can clearly be seen that it has a multi-faceted nature which is not easy to define in a unified definition. Common to these definitions are however notions of confidence, belief, faith, hope, expectation, dependence, and reliance on the integrity, ability, or character of a person or thing [7].

[1] http://www.thefreedictionary.com/trust

Trust is used in our daily lives in interaction where partial information is available [7]. We assume that a person will act as expected. Yet trust is elusive and it defies stringent definitions. It also is largely invisible and implicit in society [7]. A person may take trust decisions on the evidence available, however trust is situation specific and asymmetrical in nature.

An attempt at a definition is made by Gambetta in [28]:

> Trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such action (or independently of his capacity ever to be able to monitor it) and in a context in which it affects his own action.

Note the word ”subjective” in this definition which again hints at the absence one.

A useful classification of trust which divides it into three modes is given by Zucker [60]:

> There are three central modes of trust production, each with associated measures:
>
> **Process-based** Where trust is tied to past or expected exchange such as in reputation or gift-exchange
>
> **Characteristics-based** Where trust is tied to a person, depending on characteristics such as family background or ethnicity
>
> **Institutional-based** Where trust is tied to formal societal structures, depending on individual or firm-specific attributes (e.g. certification as an accountant) or on intermediary mechanisms (e.g., use of escrow accounts)

Interestingly enough, all three of these modes are somehow related to trust management. Although the characteristics-based mode is obviously the more important one.

The process-based mode can be seen as a mode of reputation. As past experiences are important in the decision process to trust or distrust, it also will play a part in a human-based trust management system. This goes as well for institutional-trust, although this is more an indirect effect. It might be more appropriate to say that this last mode of trust is more profound in traditional trust management systems which are certificate-based and therefore trust is expressed in the certification authority. Apart from this, institutional- or system-trust may be relevant for identity-management.

Trust has dynamic properties as it is self-preserving and self-amplifying: it increases through successive positive interactions and degrades through misuse, disuse and/or abuse [7]. Another dispositional aspect is the measure of propensity (tendency) to believe in others. The word measure may be very misleading as it is not clear how this should be measured at all. As all trust is dependent on situation and available evidence, no two trust-decisions are similar and the tendency of trusting people may not be measurable at all.

2.4.2 What is Reputation

In the previous section we have seen that trust is dependent amongst other things on situation, context and available evidence. However, again as with trust, there seems to be no clear definition here as well. Again we are stuck with vague terms and description.

When we look at the dictionary, [1] the following descriptive definitions come up:

[1] http://www.thefreedictionary.com/reputation

- The general estimation in which a person is held by the public.
- The state or situation of being held in high esteem.
- A specific characteristic or trait ascribed to a person or thing: a reputation for courtesy.
- The opinion generally held of a person or thing
- A high opinion generally held about a person or thing
- Notoriety or fame, especially. for some specified characteristic (Latin reputation)

Reputation is used in various digital systems as can be seen in [44, 51]. If we further define reputation as the expected way of behaviour of a certain entity given a certain set of evidence, we could argue that reputation as such is a basis on which a trust decision can be made. In this respect reputation and trust management are closely related [7].

2.4.3 Risk Management and PCT

There is one aspect of this story that is not yet explained but does play an important role in everyday life and maybe even in the whole story of trust management. This aspect is, that trust and risk are inexorably linked to each other [7]. It may or may not come as a total surprise that this is so. Similarly as with trust it is part of our everyday lives and we take risks therefore all the time. Let us illustrate this with a couple of general examples before dragging PCT into this one as well.

Consider our example from section 2.3.1 in which we sketched a driver turning into the main square at the Capgemini building. We trust him or here to drive into the square correctly. Why? Well because we assume that the person has a drivers licence and therefore is qualified to do so. In other words, we have institutional trust (see 2.4.1) in the driver having obtained his licences fairly and therefore is able to drive the car correctly.

Well yes, but what if... That is indeed the point we are trying to make. What if say the mobile phone of the driver rang and distracted the driver. He or she could still hit the infamous sitepost of the entrance. We did just exclaim our complete trust in the drivers ability to execute the turn correctly right? As it is this trust may be well founded, it is not foolproof. Distractions can and will happen. Therefore there is a certain amount of risk involved that the driver will not execute his turn correctly.

More generally it can be said that where trust is involved, even if that trust is in very common things where you would not directly think of it as trust, risk is the flip side of the coin. You trust that when you get home it is still there, there's power and gas available for you and that all your belongings are still there. However, there is always that bit of risk that this one day will not be true at all [1].

Now let us see what this means in terms of PCT. We will do this by example and reference to two article specifically written about risk management and PCT. Let us consider the example given in [38] where a risk manager needs to decide if he is going to move an entire storage facility from the museum he works for or not. This is given the fact that nearby the building an old bomb, left over from the second world war, is going to be dismantled with the chance that it still might detonate after all these years. The tradeoff that must be made in this case

[1] If we believe Murphy's law the chance of this happening is 100% as the law states: anything that can go wrong, will go wrong at the most inconvenient time

was moving the entire inventory with the certainty that not all pieces would survive intact or the change, upon detonation, that the entire inventory would be lost [1]. We will tell you what he decided later and if it was the right decision.

Risk seems to be measured by the likelihood a future event is occurring with a certain amount of harm [38]. risk assessment is mostly done on information base upon past experience, own gut feeling and peer group experiences. Three parameters are central to risk management, they are: thread, Impact and Likelihood [37, 38]. The thread factor is taken as given whilst impact and likelihood define the seriousness of the given thread. In other words: Impact (I) + Likelihood (L) = Risk (Perception) (R) [37].

There are five possible courses of action (although often only four of them are listed) [37]:

Ignore: This one is often seen as similar to acceptance, but accepting implies that the risk factor has been given some consideration at least. When ignoring risk this is not been done. Whether or not to list this option is debatable.

Avoid: Avoiding involves future action and possible not taking some actions to get rid of its possible consequences i.e. not going into a certain transaction may avoid the financial risk if it fails.

Transfer: Many think that when they outsource part of their business they also transfer the risk associated with it, however form their own company perspective this is incorrect. The value of a company is calculated by its total value minus the value of risk not managed or transferred.

Accept: As opposed to our first option here indeed risk evaluation is done

Reduce: Risk is never an isolated effect and can be divided in three stage: before it happens, when it happens and after it happens. Reducing risk involves five stages with accompanying counter measures (see the table below).

The evaluation done before we can accept risk is a 2-way process:

1. The potential impact is too low and action is not really required
2. It is impossible to come up with a reasonable counter measure

The likelihood factor is not important in the first option as there are lots of small and acceptable risks in a company like employees taking home pens, making private calls on work phones and using other company assets like printers for private use. The second option however can be potentially disastrous. All examples that can be given for this option have two things in common: the potential impact is very high and they are outside your span of control [37]. The initial risk needs to be accepted, but not the consequences associated with it.

In this table we list the five risk stages and there accompanying counter measures required for risk reduction [37].

Stage	Counter measure
Threat	Preventative measures
Incident	Detective measures
Damage	Repressive measures
Recover	Corrective measures
Evaluate and correct	Educational and corrective measures

[1] Note that we are talking about risk management as a convenient wrapper to illustrate what risk is

Risk management essentially is about planning our actions for a future event that if it occurs may cause harm to our company. But, according to PCT, this is impossible. As we have explained in section 2.3.4, behaviour/action result automatically from the comparison between reference and perception. The feeling of being in danger, physically or in the function of risk manager, is defined in PCT as an error signal which needs to be resolved [14, 37, 38].

Back to our example. The risk manager of the museum decided not to move the inventory. Probably the costs or any other factor weight heavier than the thread of losing the entire collection. As we define this with the help of PCT we say that an internal goal (reference) must be set. The goal here would have been that the only purpose of a museum is to collect artefacts and preserve them for future generations. Based upon this goal, the risk manager made a very strange decision. Granted there was not much information available about world war II bombs an d the lack of information may have caused him to take no action. Also if he had experienced a disaster before he might have acted differently as well [38]. But as it is, he did not experience any error signal about this decision. It played out alright in the end which is certainly not always the case. As bizarre twist to the story and interestingly enough, the decision not to move was partly based on his experience moving household items and not moving valuable inventory by professional movers [38].

Risk and trust are each others opposites. We trust that something positive will happen, yet we risk it going wrong at the same time. There is a trade-off point on which the trust-part is stronger than the risk of it going wrong. However, as this is outside the scope of this paper, we will not explain this in detail.

2.4.4 Defining Trust Using PCT

In section 2.4.1 and 2.4.2 we have talked about trust and reputation separately. However, we did mention that reputation is a building block for making trust decision at the end of section 2.4.2.

This point is illustrated in the book ”Freedom From Stress” by Ed Ford [12]. In his book Ed Ford applies PCT to marriage counselling where his recipe for fixing a broken mirage is based on the concept of quality time. That is, spending time together with mutual awareness which builds up to a sense that the other is a worthwhile person. This again leads to friendship, trust and even love. This, of course, only works if both parties are willing. This concept of quality time can be viewed as a concept of reputation where spending time together leads to a more informed picture about the other person and there more evidence is available to build a reputation. As the concept of quality time leads to trust, we can safely conclude that reputation, if it is a positive one, leads to trust, or at least is a solid part of its foundation.

Philip Runkel, in his book ”People as living things” [49], takes a very different approach when he defines trust as:

> I might trust you to wallop my head the moment my back is turned. I use the word only in the positive sense, that I trust you not to bash my head in, that I believe you will act for my welfare or at least will refrain from acting to my detriment.

It seems more likely that he actually is defining the fact that the risk of the other party wanting to bash his head in is at this moment slightly less, or at least so he believes, then the fact the other party will refrain form doing so. With this statement Runkel actually acknowledges the fact that there is a very strong relationship between trust and risk.

Runkel further defines the measurement of trust, high-level, strong or lasting, as a longer duration of the period people refrain from harming the other person and/or having the well-

being of that person in mind. This is one of the prerequisites for communication and maybe extended to other sorts of circumstances or simply to continue for a longer period of time or both [49]. When trust is low however, precise and detailed specification of goals may be requested which can be sued as checklist.

Runkel further defines that: ”Communication and trust are reciprocal”. That is, a little of one of the two encourages the other and is even necessary. However one can outrun the other if a person is set to persuade a second person to take an action that furthers the goals of the first person.

Runkel also links concepts of cooperation and coordination to the principle of trust. He give two examples of this by explaining about a bus ride and the mutual goals he has similar to those of the bus driver, which are low in the PCT hierarchy. The other example explains about a fire engine with two sets of steerable wheels and two drivers coordinating the movement of the truck. In both cases, some form of trust is involved.

Cooperation in this perspective is seen as not just wanting to reach the same end. It is meant so that both persons cooperating actually must want to enter into a joint effort to reach the same end. Or in PCT terms; the cooperation itself must become an internal goal or reference so that both persons act to perceive that they are indeed maintaining the cooperative relationship [49]. Bill Powers, in an e-mail conversation I had with him, he looks at the trust problem as a set of evidence that would be consider sufficient to convince the other part, person or institution, that you are trustworthy. Additionally, do we provide the evidence ourselves and if so do we hide counter examples of our trustworthiness or not.

What this basically comes down to is if one’s own perceptions are the true representation of reality or not. This considering the inherent conflict between what we want reality to be and what we can prove is true. Powers states that we never can look directly at reality without relying on our own perceptual systems, therefore any proof has to be indirect, statistical and relative as we judge some perceptions in terms of other perceptions. Powers also adds upon our statement that trust is based on past experiences that it also depends on how we analyse them.

A PCT related description can be given by not directly looking at evidence or statistics, but to try and find out what the reference conditions are of the other person. Which is more in the lines of how we are organised as living control systems. This a more model-based approach to the problem. Powers states that the main question here is: ”whether the promised behaviour is consistent with what I understand the other person to want”.

Powers ends the mail with the following:

> Perhaps what this comes down to is simply an analysis of the ways people find out whether they will benefit by trusting others, or by being trustworthy themselves – or else by doing the opposite.

In the context of PCT we would say that trusting or being trustworthy must ”benefit” some internal reference. That is, it perception created by trustworthy behaviour must match some internal reference. People seem to have varying motives (higher-level reasons) to trust or wanting to be trusted. Powers states that generalising and finding some common characteristics is probably a mistake [1] [2].

In a follow-up message to the one in appendix A Bill Powers poses the question if there even is a solution to the issue of ”digital” trust that guarantees 100% security. This, in respect to this

[1] Unfortunately for us, that is exactly what we need for a definition of digital trust in this paper

[2] The entire mail can be read in appendix A

field of research, is a rhetorical question to which the answer obviously is ”no”. Powers solution is a strategy in which the risk for both parties is equalised such that either party will experience an unacceptable loss if trust is betrayed and of course satisfactory gain if it is justified. If a proposal of such kind is rejected, this is a pretty good indication that the rejecting party can not be trusted.

The solution Powers proposes is based on the concept of the ”prisoner’s dilemma” or a variation or the opposite of it [1]. If both prisoners stay silent and therefore not betray the other person, they both gain. However if the solely care about themselves and therefore betray their fellow inmate they both will receive intermediate sentences.

The prisoner’s dilemma is created by having a third party who is able to set and enforce the rules of the game. There is no choice of not playing the game, The rules could bend though. For the prisoners this would mean testifying in such a way that there testimony later would be invalidated and they would establish themselves as a liar or an incompetent observer. They followed the rules in that they could not be accused of staying silent. However, what this ultimately will bring them is unclear and probably dependent on the rules or social context dealing with incompetent observers and liars [2]. The question of trust in this respect is about who makes the rules for this game and how are they enforced. If a third party that is empowered to do so is brought into the relationship, it becomes possible to make sure that a person who betrays trust will suffer yet both will benefit equally indeed both parties remain trustworthy.

As we define this in the context of PCT we say that this third party must be someone or some institution that both parties perceive as being superior to them. In other words: there is some system concept that both parties must adopt that is accepted as superior to principles like ”me first”. A person that does not accept this kind of concept arrangement will therefore not enter in to such an agreement as the perception of putting trustworthy behaviour above the internal reference concept of ”me first” will create a sufficiently large error signal. Powers describes this as the origins of government and law. This system however will only work if and only if there is a third party involved that is 100% trustworthy or at least significantly more trustworthy than the average participant in an agreement. [3].

In the introduction of this section we proposed that it would be an easy way out to simply define ”a list of controlled variables” to define trust in PCT. However, it seems that even that is as elusive as a definition for trust itself. Even with a theory that defines in clear details the working of any ”living” human control system, it seems that a subjective and probably emotion-related concept of trust is undefinable. Sure we have given a lot of descriptive explanations and we obviously can pinpoint which parts of the PCT hierarchy are involved with the trust concept, but a pure and simple definition explained by one of many control-loops each with its own controlled variable (cv), is simply impossible.

2.4.5 Conclusion

In this section we have given various descriptive definitions of trust, reputation and risk, by example of risk management. We have also given the visions on PCT and trust by two of the

[1] Prisoner’s dilemma: both prisoners are asked to testify against each other, if they do they gain freedom. If both testify they both get an intermediate sentence, if one testifies and the other stays silent he will get set free and the other will get a long sentence and if both stay silent they both get a short sentence

[2] Note that in the context of PCT the term observer or observation could never be combined with incompetence. We would define an ”incompetent observer” in PCT we would need to define him as a person that has an internal reference set such that the resulting error signal and the automatically following behaviour is incompetent in the perception of the other party

[3] The entire mail can be read in appendix B

foremost representatives of the theory. However, the only definition of trust we can come up with is still a descriptive one at best. In the context of the research questions posed in this paper, it is worthwhile to note that reputation is a basis for trust if it is positive. This statement can be used to later define a digital trust definition based on available evidence acquired from observation and recommendation. Another useful point was made by Bill Powers in mentioning the use of the ”prisoner’s dilemma” to resolve part of the trust decision. This can be very useful in situation where trust is low or not enough evidence is available to come to a reasonable trust decision.

Philip Runkel in his book ”people as living things” [49] talks about his encounter with a group of mountaineers who did not talk so much about the challenges or the thrill of hanging at the end of a rope and dangling in mid-air. They mostly talked about the deep feelings of trust, affection, and camaraderie that developed between the team members. Runkel concludes this by stating that this feeling of mutual commitment and joy, whatever the task, is what is meant by esprit de corps and is also verily close the one of the meanings of love [49].

With this last statement we can place trust somewhere in the hierarchy of the living control system along with our other feelings and emotions as described in section 2.1.4. This may as well be precisely the reason why it is so hard to precisely define trust. One particular statement in the mail exchange with Bill Powers, which we already mentioned in the previous section, sums this up quite nicely as (see appendix A):

> People have varying motives (higher-level reasons) for wanting to be trusted, or to trust others. It’s probably a mistake to generalise and try to find some characteristic that everyone has.

Trust may be elusive to some, with the help of PCT we can at least pinpoint where in our personal hierarchy of control we can find the control loops that combined on the conceptual levels will form the feeling of being trusted and the behaviour of trusting others. That this in the end is based upon past experiences and the personal way of analysing them is directly captured in the way the reference signals in the hierarchy of control are constructed. If they happen to mismatch the current perceptual signals we probably feel betrayed.

2.5 Behaviour Controls Perception

The title of this section refers to three words in the subtitle of this chapter which are ”Behaviour Controls Perception”. It is all fine and well to quote others, come up with some definitions, but as this theory essentially is a revolutionary way of looking at human behaviour, I wanted to include my personal views on the matter in this paper as well. I was inspired to do so by an article entitled Three ”dangerous” words by Thomas Bourbon [6] which was originally drafted as a possible foreword for the book ”Making Sense of Behaviour - The Meaning of Control” by Bill Powers.

It is probably a weird thing to see a theory from the social science playing a prominent role in a paper about a computer science topic like trust management. But in my opinion that is exactly what is missing from this field of research. As you may see in the next chapter, a lot of research has recently been done in trust management. Apart from one project however, nearly none of it really considers the ”trust” factor in these systems. It seems that either this is too difficult or that people involved in computer science research do indeed look no further then the digital boundary of research. I think it is important to our field of science to include such theories as this in our research to better integrate our findings in a more expending digital world.

There are a lot of theories about human behaviour, most of them are probably more accepted in today's social sciences community than this one is. The original article which inspired my to write this section [6] illustrates this. I must admit that at first I was sceptical about using a theory from the social sciences in a technical research setting. But the more I read about the theory, the more I came to realise that it is indeed an explanation of how we human beings work. And if I wanted to define something as complicated and probably elusive as trust, I would be better of with a theory like this than any other half-true definition I could find in the dictionary.

I will not defend this theory against any others here, I have already done this to the best of my technical ability and background in section 2.3. However, it is not only the fact that PCT does indeed explain human behaviour in a way that seems completely fitting the truth that it is an interesting theory when dealing with trust management. As already pointed out in section 2.1, the theory does have a technical background as well. Not in computer science however, but electrical engineering. As the two fields are very intimately linked, it is a firm basis we can use.

I believe that for all intends and purposes PCT does have a future in computer science. Maybe not everywhere, but certainly in parts where human elements are, or should be, part of the equation. Not only research in trust management can, and with this paper hopefully will, benefit form this approach, other fields often bundled under the term ”human computer interaction” would well realise that as specially there the human perception is the most important one to take notice of when designing interfaces. It is that perception that through experience and internal reference lead to the users actions. Maybe, without noticing it and naming it intuitive interface, they are already applying PCT to their designs. If so, would it not be more effective if they indeed would also perceive doing so?

I encourage you, the reader, to spend some time and read the available literature on this topic and decide, with an open mind, for yourself. As Bourbon shows in his article [6] a lot of people were convinced that PCT is completely correct in what it is proposing. It however simply did not fit their career plans to change theories. Let their loss and mistake not be yours.

If you have read this chapter and understood what PCT is trying to tell you, you will also understand why ”behaviour controls perception” and why it is not the other way round. Even so, if you understand PCT you also understand its usefulness.

Chapter 3

Trust Management: A Literature Study

In this chapter we will give an overview of what kind of trust management research has been done so far. This chapter is by no means an exhaustive representation of what is already done in this field. Parts of this prior research will contain interesting starting points for the system we are designing. However, the majority of the research done leaves a lot to be desired.

Most research that is named to be in ”trust management” actually is not, at least not by our definition of trust. Most common reason is the exclusive usage of certificates for authentication as well as for credentials. Trust here is expressed in the certifying party and not the certificate holder. In general these are not the same. Other reasons lay in the fact that there is no clear definition of what kind of trust is used or even that the term in itself is used correctly. Most of the time the term ”control” would be a better fit for the situation. Note that this does not necessarily mean similar to the definition of control we gave in 2.1.1. The research done in Pervasive Computing goes a long way in the right direction. However, this is more because of the similarities in the type of system than because of the trust management designs themselves.

The only project that was found that may provide a starting solution for our design is the SECURE project, defined at Trinity College Dublin (TCD) between 2002 and 2004. It uses a definition of human-based trust in its trust management framework. Although the underlying theory may be different to the one we are using, it is so far the only project at least trying to define trust in a human-centric way.

In the next section we will discus the early work in this field which was mostly done by Matt Blaze in the 2nd half of the 1990’s. After that we will discuss other materials on trust management that were done over the years, but do not fit into the other sections of this chapter. Then we will discuss the field of pervasive computing and what it has to do with Jericho. After that we focus on one particular project in the Pervasive Computing field called the SECURE project. We end with a summary of pros and cons of these topics and their relevance for the requirements which will be defined the next chapter.

3.1 Early Works

In this section we will discus some early works in the field of trust management their general principles. Most early work was done by Matt Blaze resulting in two different systems called PolicyMaker and KeyNote which will be discussed later.

3.1.1 General Principles

Matt Blaze describes trust management as a system that primarily deals with privileges defined in a programming language. This opposed to traditional methods of access control which involve an authorisation decision by the system [5] based on authentication and/or access control lists (ACLs).

These kinds of changes are necessary for scalability and decentralisation issues. It is claimed that ACLs are not suitable in these cases which is correct. In distributed systems ACLs need to be updated for every single host. [1]

The trust management systems proposed in early research all work on the basis of certification. If someone has the right certificates, matching a certain security policy, he or she is allowed access. The trust relation is therefore based on the certificate and the person/system delegating access credentials rather than on the person actually using the certificate. This model might work for distributed systems and maybe even for company domains, but it will not work on a global scale. This is simply because it still would need an authorisation system to verify the keys with which the certificates were signed. This could be done by using the GPG/PGP key-system, but the question of scalability remains. Add to this the fact that these systems are limited in their capabilities in that a single key can be seen as being trustworthy even if only one person actually declared so and the reader will agree that this is no basis of trust.

Also PGP, and x.509 as well, are very specific system. They do define part of the trust management principles but often in a very narrow way.[4].

3.1.2 PolicyMaker And KeyNote

In this section we will look at the two early systems by Matt Blaze et al. called PolicyMaker and KeyNote.

PolicyMaker was the first system that handled the authorisation decision directly instead of leaving it indirectly with authentication- and access control services. It also provided a proof of compliance in an application-independent system. The system was made so it could handle signed requests, credentials and policies.

KeyNote was designed along the same principles as PolicyMaker was. However, two additional design goals were added namely: ease of integration into application and standardisation of the system. To achieve this, more responsibilities are given to the KeyNote system. This as opposed to PolicyMaker where the calling application is given more responsibilities than the trust management engine itself.

Both systems have an internal compliance-checker that needs to calculate the actual trust decision. The information both of them receive is practically the same, the way they calculate is not [5]. Credentials, policies and requests are specified in certificates. Although this is similar for both systems, they way certificates are handled is different. The underlying trust relationship however is trust in the certificate and not the actual user.

In the following paragraphs we will describe aspects of both systems. We will point out their shortcomings, where the trust relation is situated and, if applicable, any positive points.

[1] An ACL is a static list of permissions either allowing or denying access on the basis of various information including username/password authentication, IP-addresses or certificates.

The system works by evaluating policy assertions. It only does the evaluation and leaves the collection and cryptographic evaluation of the credentials to the calling program. Credentials and policies are cryptographically signed by the public key of the issuer. These signatures are, as said, evaluated by the calling program and not by PolicyMaker. The same holds for the policies needed for the ”trust” calculation.

In general this puts the trust relationship not in the place where it should be. It puts it right in the relationship between PolicyMaker and the calling program and even then the wrong way round. PolicyMaker implicitly trusts the calling program to do the cryptographic evaluation and do it correctly. However, no checks are in place to make sure this actually happens. Hacking this is even made easier by the fact that credential gathering as well is left to the calling application. The designers claim that both of these facts, both tasks left to the calling application, are positive feature [5] section 2.2. However, for obvious security reasons this will not hold.

The system does try to protect itself by stating that any assertions [1] needs to be written in a programming language that can be executed and evaluated safely. For this purpose a safe version of AWK was developed.

All assertions use a write-only data structure to collectively create the required proof. It is described as some kind of blackboard on which the proof is constructed. The reason that all assertions can access all written data, or so it seems, is exactly the reason why one faulty assertion will be able to overwrite or ”hack” the proof process. The only hope the system has in avoiding this is the fact that PolicyMaker does check if assertions behave in a cooperative manner. If not they will be discarded. The other decisions required are in what order to run the assertions and how many times to run them. However, this purely is an implementation detail and clear guidelines on the latter point nor what is expected to be ”cooperative behaviour” is defined.

The most general version of the compliance-checking problem, which is not implemented in PolicyMaker, well accept any arbitrary function. However, the version that is in PolicyMaker only accepts those that are monotonic, [2] which has the benefit of being analysable and provable correct on a well-defined set of assertions. Site-affect is that it will fail on negative credentials or revocation-lists which are used in practical situations. It is possible, mostly, to rewrite negative credentials to positive once, i.e. not being on a revocation list maybe rewritten to the fact that a certificate of non-revocation is presented. The challenge still remains how to handle revocation of credentials at all and this trade-off does not make it any easier.

Finally it is stated that by using monotonic assertions a conservative approach is taken on security and that dangerous actions are only allowed if an adequate number of positive assertions is presented. This because dangerous actions are disallowed by default. What a ”dangerous” action actually is, is not specified. More information about PolicyMaker can be found in [4].

[1] An assertion is the combination of policies and credentials denoted by the tuple (f,s) were f is a program describing the authority being granted and the parties it is granted to and s denotes the source of authority

[2] Basically: if a monotonic assertion approves action A when given evidence set E, then it will also approve action A when given an evidence set that contains E.

KeyNote

As already stated before, the basis of both systems is their similarity in using credentials for authorisation. However, KeyNote clearly seems to have less concerns security-wise than PolicyMaker has.

KeyNote handles all cryptographic checks itself. This is also were they should be done. It means that in this system no implicit trust is present between the KeyNote system and the calling application. KeyNote therefore is no longer dependent on the calling application to do its job correctly. Also in KeyNote all assertions and credentials are written in one specifically chosen language to extend the inter-operability of the system and force carefully written credentials that work smoothly with its compliance-checker. Note that carefully here does certainly not mean that the credentials themselves are carefully designed, it just means that their construction is. Their meaning, and therefore the access rights that are allowed, may be completely stupendous and insecure.

On another level the implicit trust in the calling application still exists. The calling application still is made responsible for gathering the credentials, requesters' public keys and other relevant data. The remainder of the info gathered by the calling application is put into a Unix-like shell environment constructed with key=value pairs containing all information deemed relevant for the request and necessary for the trust decision. Although it is an improvement, too much "trust" is still put into the calling application behaving "correctly". However, this is more relevant to the information put in the environment that the signed data [1].

3.1.3 Summary

In respect to the basic way of working, both systems are equal. That is in the respect that both are certificate-based, both have a compliance-checker and both rely in on the information given to them by the calling application. The format of assertion-notation differs between the two implementations. In PolicyMaker policies are local and therefore trusted automatically, in KeyNote this is not the case. This also holds for where the responsibility is placed for the cryptographic signature checking. PolicyMaker relies on the calling application whilst KeyNote handles this task itself.

It is clear that KeyNote is a more advanced version of PolicyMaker, not only by the fact that they were designed by roughly the same authors. In its design decisions, probably because of criticism on the earlier system, KeyNote seems to be more secure and more conscious of possible threads.

In the case of usefulness within the Jericho context they both fail. This partly because too much is left to the calling application. Both systems, apart from the cryptographic checking, are not much more than compliance-checkers calculating their result solely based on the information provided by the calling application. In the Jericho system we need the trust broker, and therefore the trust management system, to be able to decide not only based on the information provided by the user, but also on reputation data and other sources of information.

[1] Note that nowhere is stated that the data inside this environment is encrypted or even signed at all. It therefore is my assumption that it is not

3.2 Other Trust Management Research

In this section we discus a couple of papers that do not fit into the Blaze-category nor in the field of pervasive computing which is discussed in the next section. In this section we first will discus a related topic to trust management, reputation systems. This is followed by another implementation of trust in distributed systems. We end with a discussion on trust management in communication networks.

3.2.1 Reputation Systems

In relationships between human beings, reputation plays a major role. In fact, the more you know about the person in question may play a significant role in your decision to trust someone or not. This is easily accomplished in an offline world, you normally meet the person in question in real life. Online however, this is next to impossible or at least very difficult.

In a natural way, a reputation is build up over a long-term relationships. This is a twofold process: firstly the history of past interactions is informative about the other persons abilities and dispositions, secondly there is an incentive for good behaviour because of the fact that this may have repercussions on future dealings [44].

To achieve this for an online reputation system, the gathering of feedback from the users of the reputation system is essential. In the natural way of course feedback is constructed by observations and perception of the party who's reputation you are constructing. Online this is much more difficult because of the short-time and ad-hoc relationships generally part of digital transactions. Examples of online reputation systems, although limited in their capabilities, are those of eBay, Amazon and Bizrate [44]. In the case of eBay it is a simple integer value that can hold 1, 0 or -1 for positive, neutral or negative feedback and the aggregation of this feedback is a simple summation [51] section 3.1.

Reputation systems have at least three properties [44] to be effective:

- Long-lived entities that inspire an expectation of future interaction
- Capture and distribution of feedback about current interactions (such information must be visible in the future)
- Use of feedback to guide trust decisions

Apart from this, some significant challenges remain, namely eliciting, distributing, and aggregating feedback.

Eliciting feedback has three related problems [44]:

- People may not be bothered to provide feedback
- Eliciting negative feedback is very difficult
- Difficulty of getting honest reports/feedback

In systems like eBay there is no incentive to provide feedback, even if it is only a couple of mouse clicks, people will tend to forget or simply do not want to be bothered by it.

Other problems lay in the distribution of feedback, These can be found in the are of portability of feedback data to other systems which makes the information limited in its effectiveness.

Another factor is the numerical format of the feedback in current reputation systems. This is also a very large aggregation issue.

Another problem is the usage of pseudonyms which can be changed. The trouble is that these can be changed, most of the times. The effect of this is that the person behind these pseudonym may start with a new and cleared reputation. This because it is near impossible to link the new pseudonym to the old reputation data. The only solution for this is to disallow pseudonym changes, the so-called once-in-a-lifetime pseudonym.

Reputation of an online user can and should be part of the trust decision. Where this is absent for new, and therefore unknown, entities, a default neutral value should be imposed. In the trust management system we are deciding, reputation can be based on several forms of evidence. This can range from the hardware a person is using, to the location and specific times of entry/exit or other measurable behaviour of the user.

Some different models and calculation schemes are provided in [51]. Although the mathematical nature of most of these places this outside the scope of this page. Although the basic notion of reputation will be part of the design, the mathematical repercussions and proofs are left as point of future research. A good starting point for that may be found in [51].

3.2.2 Decentralised Trust Management

In this paragraph we discuss a multi-layered model that is concerned with detecting malicious network usage and its effects on long-term trust relationships. This is done in the setting of federated systems over wide-area networks and the accompanying research and test-bed setups [1]. It must be noted that whilst not all such traffic is malicious in nature, most of the anomalies found are often irresponsible or even naive. Because of this, full accountability is required. Generally holds however that eliminating all false positives or negatives is not necessary. Reasons for this are that signatures of past incidents are easy to detect if such an incident occurs again and the background noise created by real threads on the Internet cause enough of a diversion which makes it impossible to devote resources to respond to all anomalies detected, so only the most significant ones will be responded to.

The following three layers are defined [10]:

- Authentication and authorisation layer
- Accountability layer
- Anomaly detection layer

In the remainder of this paragraph we will shortly discuss each layer [2]. Note that the first two layers together actually form the triple-a framework common in security practise [3].

[1] Federated systems make it possible to access remote resources over the network spanning multiple administrative domains wit the possibility of malicious use like denial-of-service attacks.

[2] The system described here was designed with PlanetLab in mind, which is an open, shared overlay on the Internet for developing global-scale network services. As these services are experimental, programming errors, naive service design and analysis or irresponsible use of resources are major contributors to network traffic anomalies

[3] AAA-framework: authentication, authorisation and accountability [59, 58]

Authentication and Authorisation

As per default in distributed systems, it is impossible to let one centralised entity be responsible for every authentication request. It is therefore necessary to distribute the trust management structure to allow it to be flexible enough. The resulting systems will have mechanisms to allow it to express, delegate, and verify trust relationships by using public-key cryptography based on fully accountable paths of trust [10]. Note that accountability itself is part of the next layer.

The authentication mechanism in this system uses a set of ”unforgivable” tokens which hold a random very large number, an object identifier and a set of actions that can be performed on that object. Obviously the object referred to is the one the identifier is of and the one that is being requested access to. Delegation is handled simply by the delegating party sending the associated capabilities to person these are delegated to. Key advantage is that no additional delegation systems are required [10].

Three main issues are present in all large federated systems: Firstly there is an issue with the paths of trust in this system. On a delegation chain P1 to Pn where Pn abuses his delegated access rights, knowing who Pn-1 is, is vital. This because in principle this is the person who delegated these rights to the culprit. However, there is no easy way to do this.

Secondly, systems with distributed capabilities generally presume that all communication channels are private and secure. It therefore is reliable to exchange delegation information over them. In large federated systems, and the Internet in general, this is certainly not the case. This leads to all kinds of problems including the one we are trying to answer in this paper: how to make sure with high probability that the person you are talking to is indeed the person intended. If not, you may be delegating to someone entirely different with possible disastrous consequences.

And thirdly, distributed systems may have great difficulty with some types of trust relationships up until the point that they become impossible to express. Problems related to this point are mostly concerned with locally generated access-controls which then are distributed or partial delegation schemes. In both situations the distributed nature of the system hinders the process or indeed can make it impossible to function [10].

These three issues are partially solvable by using either PolicyMaker or KeyNote in the system. Trouble is that these systems also assume the existence of secure communication channels [10]. This is exactly why these kind of systems are unusable in today’s hostile network environment.

Accountability

The accountability layer of the design offers two types of services: continuous monitoring of how the trust relationships are used and periodical logging of principal behaviour. These services together will make it possible to find the chains of identity and delegation when anomalous usage is detected.

A lot of monitoring information can be acquired on OS level. Specially in Linux/Unix systems the /proc file system and TCP-implementation provide a lot of details. However, also extra more fine-grained principles of monitoring are required for full accountability. Most of the data gathered through say the /proc file system specially for networking purposes, is aggregated. It will not provide the details needed on the basis of ports, IP-addresses and local principals which a monitoring system used for full accountability needs.

These kind of extra logging methods will create overhead, however with the current speed of processors and networks this is negligible in large federated systems. The only challenge

remaining for the monitoring is that a mapping needs to exists between local principals and global once [10].

The logging facility can be implemented with standardised logging and database technologies. The data from the monitoring service can be collected periodically from the different notes. It then has to be mapped back along chains of trust. This logging storage provides a historical and traceable of the behaviour of principals. In federated systems this information needs to be put into a persistent storage for two reasons, firstly because misuse may require correlation of data over time and resources, secondly because nodes can and will crash and it is not known how much data is required for postmortem analysis in case of misuse [10].

Anomaly Detection

This layer is the anomaly detection layer that is required to detect misuse before it can escalate and lead to external complaints. The total anomaly detection needs to be done locally per note as well as distributed [10].

For the local anomaly detection the system makes use of the logging information and accountability principles from the accountability layer. This information is then processed through a rule set to detect anomalies. Appropriate warning levels and subsequent actions are attached as well. These actions are defined per anomaly type and warning level. The anomaly detection on this level focuses on the network usage of applications, local resource usage can easily be bound wit appropriate OS controls.

In general it can be said that the anomaly detection in distributed systems is the converse of intrusion detection [10]. In intrusion detection traditionally incoming traffic is scrutinised and scanned for anomalous and potentially dangerous traffic. In the anomaly detection in federated systems, or distributed systems for that matter, just the opposite is done. To implement this a reversed intrusion detection system is required. Main Benefit of this is that existing rule sets can be used. Side-effect, is that the anomalies that are detected are on aggregated information for the entire node. The challenge therefore is to adapt the system to be able to detect anomalies on an per application basis [10].

For the distributed anomaly detection a rule set is either applied incremental or aggregated. This method of working combines the correlation of information over multiple nodes. This is needed to detect anomalous actions tat are perfectly allowed on a single node, but not combined on multiple nodes. An example of this is a port scan running on 1024 nodes each scanning about 64 ports each on the same target. If only a single node would execute this it is not a thread, however if all of them execute, possibly simultaneously, it is easily seen that this would become a major thread to the intended target.

Key challenge in this is to keep the overhead of the anomaly detection system as low as possible whilst at the same time the anomalies themselves have to be detected in a timely manner. One brute force way of doing this is having a separate host running a traditional intrusion detection system in reverse and replicating all outgoing data of all nodes to this system. Obviously for large distributed system this is not possible at all. Even if the central host could handle the load in processing power, the network bandwidth required would be such that all host together would produce less bandwidth than available at the central anomaly detection server. This has large overhead and scalability issues [10]. The only way to tackle these issues is the leverage location-independent, key-based routing (KBR). In this scheme all nodes take a fraction of the total computational and bandwidth requirements which significantly improves the scalability whilst still detecting large anomalies.

Summary

Whilst this research and the tree layers are quite interesting and possibly even useful in some parts of the Jericho vision, sadly nowhere is defined how trust relationships are achieved or even trust itself is defined or calculated. However, it can be assumed that this will go through old and proven human-to-human trust factor. As the delegation seems to be based on the hierarchical lines that exists under the users of the PlanetLab system, this is an offline mechanism and therefore outside the scope of this paper. As such, this is a luxury that is not bestowed upon the users of de-perimeterised networks. The anomaly detection scheme discussed above may eventually be interesting in network security but will not be of further use in this paper.

3.2.3 Trust Management in Communication Networks

All trust relationship are context dependent. It as well is a complex concept that is hard to define. One definition is given by: "Trust is the extent to which one party is willing to depend on something or somebody in a given situation with a feeling of relative security, even though negative consequences are possible", [35]. Three dimensions can be defined: the trust origin or truster, the trust relationship or purpose and the trust target or trustee. Additional trust values can be added to all three dimensions.

In [36] they define a three-layer trust model:

layer I: This layer is responsible for authentication, authorisation, and policy definition

layer II: This layer provides two major services, first it provides a continuous monitoring of the usage of trust relationships and secondly it provides a periodical logging of monitoring data to provide important information

layer III: This is the automatic anomaly detection layer.

The model provided above is similar to the one we discussed in the previous section in detail. A lot of the remaining issues to trust management that are discussed in [36] are already discussed in this paper. We therefore will leave them out in this section.

3.3 Pervasive Computing

Pervasive computing, amongst other things also called ubiquitous computing, is the field of research concerned with the integration of computing devices into our everyday lives. As opposed to the desktop paradigm, pervasive computing does not deal with a single device where a single user is consciously using it for a specific task, but deals with multiple devices simultaneously employed in ordinary activities. The user of a pervasive system may not even be aware of the fact that he or she is actually using multiple systems at the same time at all.

In general it is the exact opposite of virtual reality in which the user is put in a computer generated environment. In pervasive computing the computer itself is put, and integrated into, the world outside. This field of research offers some interesting design issues [48] which are not that dissimilar of the ones found in Jericho, not only for the part of trust management.

In this section we will look at the trust management aspects of pervasive computing. We start with drawing the similarities between pervasive computing and the Jericho architecture. Following this we describe some of the work in this field on identity- and trust management. Finally we wrap things up with a short recap.

The closely related, and already mentioned, SECURE project is discussed in the next section. This project is part of the pervasive computing field, however for obvious reasons, that will be come clear in that section, we have chosen to discuss is separately.

3.3.1 Jericho IS Pervasive Computing

The reader will agree that this is a very bold statement to say the least and on several points it will not hold, however in the more important issues within the scope of this paper it will. This section explains why.

In section 1.2 we already stated that commandment 8 is the most important and that commandments 6 and 7 are closely related to the scope of this paper. With the help of these three commandments we will draw the comparison between Jericho and pervasive computing.

Commandment 8 states that authentication, authorisation and accountability must operate outside your area of control. There is no direct ”area of control” within pervasive computing. As said, computing devices are integrated into our everyday lives everywhere and we may not even be aware of using such system. It is near impossible to define what your ”area of control” would be. Maybe your own home, but even then it is possible you are using pervasive mobile services, like news and weather information, that are not under your control. In this respect pervasive computing goes even farther than the Jericho architecture itself. Therefore, Jericho commandment 8 is only defining a subset of the same issue within pervasive computing. All solution given in this field are usable in Jericho, but will be rather less complex.

Besides this both architectures have a lot of privacy issues. Commandment 8 states that there must be one instance of an identity. Privacy issues however would allow for multiple instances and/or one instance with multiple facets. In pervasive computing the privacy issues remain, however the scope is slightly different. This is because there already are multiple instances, a user may have multiple mobile devices, and the combination of the information gathered from these instances poses the privacy issues. These issues are that a lot of data combined may provide insight into a users behaviour even may lead to disclosure of his or her real-world identity. [1] Although information of a single instance poses these risks as well.

In general it can be said that most points related to this commandment overlap and that in some cases pervasive computing supersedes Jericho. Jericho only differs in its mention of a specific ”area of control” which is absent from pervasive computing.

Commandment six states that all devices must have defined a set of transparent trust levels. Where trust here is not trust as such but more a mutual understanding between contracting parties to accomplish a transaction and the obligations of each of the contracting parties that goes with it. The two systems differ only in the point that in pervasive computing we do not talk about organisations. The system is more based on people and devices, entities in short and the connecting infrastructure. Although this last part is mostly ad-hoc, wireless based and therefore as infrastructure as such difficult to define.

The fact that trust levels may vary per location, transaction, risk assessment and user role is certainly valid. As these elements constantly change in an inherently open architecture, like pervasive computing, it is strongly advisable to be aware of this and allow your trust model to be adaptive enough to deal with it accordingly.

Commandment seven states that there must be a system in place for mutual trust assurance

[1] A single device-address may be useless in that respect. Combine this with subscription data to a certain paid service and you can make the assumption that the device-id belongs to the same person.

with multiple levels. This also holds for pervasive computing. Even more so because there is no area of control and possibly even no organisation the identity can be authorised by, or at least a different one in different cases. The user centric model, central in the Jericho architecture, holds completely for pervasive computing as well. This because all the accessed service have initially, is the information provided by the user. As these possibly can be devices with low capacity as well, which may even not be network connected, the need to be able to decide if they trust the identity requesting access or not. Same holds true for the access requesting party which most certainly will have a device with limited resources.

Both commandment six and seven are part of the same group named ”the need for Trust”. This is the central theme within this paper and the an important comparison point this section is based on. All others are interesting and illustrative of the main proposition. Even the other 8 commandments will have some kind of relationship with parts of the pervasive computing scheme. This holds certainly for the fundamental commandments:

- The score and level of protection
- Pervasiveness, simplicity and ease of use of security mechanisms
- Do not assume context

Whilst the first and second point may be quite clear, just protect what you need based on the scope of the asset or risk with scalable building-blocks for your security, the third may need some explanation. [1]

Assuming context is dangerous, specially in pervasive computing where the environment may chance rapidly. This means that even a simple transition between two entities within a pervasive system may and certainly can be picked up by a third, possibly malicious, entity. Assuming context, i.e. we are in a closed room and nobody can hear our transitions, therefore can be fatal. This holds true for both Jericho and pervasive computing.

Finally commandments four and five under the heading ”surviving in a hostile world” may not need any further explanation. The usage of open and secure protocols holds for both systems as they are inherently heterogeneous in nature. That all devices need to uphold there security policies on untrusted networks is an open door we will not kick in. This as specially for pervasive computing where possibly all communication is done wirelessly and therefore inherently unsafe. Jericho is pervasive computing. Yes it is, however the Jericho architecture goes less far than the pervasive computing paradigm. Therefore, reversing the proposition to ”pervasive computing is Jericho” will fail.

3.3.2 Identity and Trust Management

In this section we will look at several papers from the field of pervasive computing that deal with identity and trust management research for pervasive computing systems. We divide this section into two parts. First we will discus some identity issues, followed by the main part on trust management.

[1] Note the use of the word ”pervasive” in commandment two meaning that security mechanisms will be their but the user may not even be aware he is using it.

Identity

In essence there are two levels of identity. The first is the corporate identity consisting of a role-based identity management or a concatenation of partial identities to simplify customer administration. The second level consists of personalised, context-aware identity management. This second level is based on location, user preferences and environment a user is in at a particular time. In this level the providing agent strives to ”control” as much information as possible to provide as good a personalised service as it can do. This will inevitably lead to privacy concerns [29].

This in total leads automatically to a third level called user-centric identity management. In this model the user is able, at least partially, to keep control over part of his personal data. It will ultimately lead to a trade-off between privacy and access rights. In one of the papers three layers of trust are distinguished: the users mobile device, the users home-server and the world around the user [29]. The accompanying framework suggests that all value-added services are done through the home base. The advantage is that the user has better control over his data and therefore his privacy. However, within the Jericho architecture this will not work. We will come back to this in the next chapter.

User-centric identity management itself however is an important keystone within the Jericho architecture. So is the possible location, context and environment of a user i.e. for providing certain extra security rights it may be required for the user to be in has or her office and nowhere else. A good method for checking this lays in the range of near field communication of which at least one example is used in the pervasive computing research which is infrared [54].

In the implementation shown here [54] infrared is used as form of pre-authentication. This means that two devices wanting to establish a connection will use this phase to agree upon a shared secret. What is useful here, specially for pervasive computing, is that no difficult and computational exhaustive methods are required. The pre-authentication is on basis of context-information and location e.g. the devices need to have line-of-sight to be able to communicate at all. This severely restricts attack-forms like man-in-the-middle and possibly even sniffing for the exchanged communication.

This form of pre-authentication may be useful in the reputation-phase of our trust management system as well. However, here it will not be used for pre-authentication but establish one more ”possibly known” fact useful for identification. A general description of the discussed above can be found in [47] chapter 10.

Trust Management

Also in this field, some research is done within the scope of asymmetric encryption coupled to credentials for access control. This is largely build on top of public key infrastructures (PKI), concerned with SPKI (Simple Public Key Infrastructure) and SDSI (Simple Distributed Security Infrastructure). However, as we already showed in this chapter, that is not the way we want to go.

At least not directly, eventually even the design in this paper will use asymmetric encryption in its processes [1] In [32] a combination is made between asymmetrical encryption technologies, specifically public key infrastructures, and trust management. It is stated that they are seen as

[1] Asymmetric encryption is not used as basis for trust or identification/authentication method. It is used solely for security purposes i.e. encrypting traffic/data and for digital signatures, not for any form of credentials.

the opposite or even duelling technologies. It is shown however that it is possible to create a hybrid PKI system [32] which uses both technologies alongside instead of opposing each other

The interesting part is that they propose to use a sort of mediator alongside the PKI system to accomplish the trust relationships required between entities that are unknown to each other. This mediator needs to have a relationship with at least one of the entities and domain specific knowledge of that entity. The specific implementation details are not important at this point and can be found in [32]. What is interesting is that this dual system seems to has some resemblance to the trust broker system within the Jericho architecture. Specifically the fact that a mediator is used with at least domain-specific knowledge of one of the connecting entities. In essence that is what a the role of a trust broker would be as well.

Contrary to the use of a hybrid PKI system, [11] states that a possible problem could be the viability of such a system to support authentication of a large number of devices present in an ever growing pervasive network. Instead, a security model based on device attributes is more likely to fulfil the security requirements.

To tackle this problem, the problem space is divided into a three-levelled model with the following levels:

User-level: This includes all human interaction in enabling or breaking security, including the user-interface

Service-level: Includes all applications and interactions with the other two levels as some sort of middleware

Infrastructure-level: Contains all hardware, computing and information resources in the pervasive network

This setup is by no means meant as a starting point for discussion about the layering of the system. It merely serves as a conceptual tool with which it is shown that achieving trust in a human-centric system will not work without careful consideration of the humans role in the system [11].

Although this is a nice way of thinking about a network, it still bares close resemblance to the old way of thinking about security: layers and protection methods per layer. This instead of protection on the data itself as is proposed in Jericho. A general description of the discussed above can be found in [47] chapter 6.

Another, again certificate-based, approach is given in [31]. Their definition of trust is based on credentials and certificates. These are used to gain access to services, smart rooms and even the coffee machine. In this paper the authors take a very interesting view on PKI's and even a more interesting one on PolicyMaker which we discussed in 3.1.2.

In PKI's, like PGP/GPG, the problem is twofold: these systems suffer from key-distribution issues and they do not deal with flexible or dynamic information like access-rights. Furthermore the system simply trusts a key if at least one other trusted entity says it can be trusted. This small basis of trust is not enough, highly subjective and non-controllable. It is possible that both entities are one and the same person [31].

PolicyMaker does bind access controls to public keys. However it does so without any authentication mechanisms whatsoever. Furthermore it suffers from the fact that non-programmers, who probably will design and implement the security policies, will have a hard time defining them. The system also is a query-engine, a powerful one, but it is not a true security system [31].

In the remainder of this paper they discuss an extended form of a public key infrastructure. This extension adds a role-based scheme to the system. At this point however, this is outside

the scope of this paper and will therefore not be discussed in detail here. See [31] for the full description.

3.3.3 Summary

In this section we have proposed that ”Jericho is pervasive computing” and we have provided proof for this based on the Jericho commandments D. Based on this proposition we have looked at various aspects of pervasive computing related to trust management and its vision on digital identity.

It is interesting to see that in this field of research the notion of a user-centric identity is used as well. Specially because we can draw upon that kind of research to define the context of user-centric identities within the Jericho architecture. It will not be a one on one match as in pervasive computing the system looks at different entities and devices instead of one single human identity, which is the way that it is used within Jericho.

On the subject of trust management, some different approaches can be seen. Most of them are still, although in various ways, based upon trust in certificates and certifying agents. So far no real trust relationships are defined between relying parties themselves. Although in essence this isn't a bad thing. We have already stated that this approach has some drawbacks, notably the scalability of key distribution schemes.

Although the implementations and design choices may differ, the main comparison between Jericho and pervasive computing holds. Although Jericho itself will be less pervasive, it certainly will benefit from any research done in pervasive computing.

3.4 The SECURE Project

In this section we will look at the SECURE project. This was a research project at Trinity College Dublin (TCD) which defined a trust management system based on the human notion of trust. As this is a comparable effort in trust management to what we are doing in this paper, it is discussed separately from the extensive discussion on pervasive computing from the previous section.

The Secure Environments for Collaboration among Ubiquitous Roaming Entities (SECURE) [7] strives to design a novel approach to security to address the problems faced in pervasive computing. In contrast to must other project SECURE does not use certificate-based credentials. Reason for this is mostly because a certificate is good at communicating identity, however it says nothing about the behaviour of the certificate holder.

SECURE uses the human notion of trusts which automatically leads to a decentralised security management approach which can cope with partial information but also holds inherent risk factors for trusting entities. Fundamentally it is this ability to be able to reason about risk and trust that allows these entities to accept risk during interaction with other entities [7].

First we will define the notions of trust and trust management used within SECURE. We follow this by discussing on using trust for security in uncertain environments. Then we have a look at their view of identity within pervasive computing. Then we look at evidence-based trust. We end this section with a short summary.

3.4.1 Trust and Trust Management Definitions

As already stated, the SECURE project uses the human notion of trust to define their trust management system. Social science uses three main types of trust [35]: interpersonal trust, system trust and dispositional trust. In this definition interpersonal trust is dependent on context. Dispositional trust however is independent of person or context. We can now state that trust is, at least partial, dependent on context. Important types of context are: location, identity, time and activity [52].

The designers recognise that trust is a useful element in any interaction when the outcome is uncertain and possibly harmful. It furthermore is stated that trust can be seen as complex predictor of future behaviour based on past evidence [52]. In essence can be said that trust allows the taking of risk in an action with possible harmful outcome.

As will be referred to a lot in this paper, trust, and therefore trust management, is at odds with privacy. The reason for this is simple, to be able to come to any type of trust decision evidence and context information are required. Even so is profiling. If the data is then linked to a real-world identity as well it becomes sensitive data and therefore privacy related [52].

In general there are two forms of trust management, namely credential-based and evidence-based. A negative aspect of credential-based trust management is that it neglects uncertainty and risk in its problem definition [30] which is a major feature of evidence-based trust management. Recent work on trust management also neglects the dynamic part of trust in the system and only focuses on the static element [7].

The project defines a secure collaboration model which is based on previously defined trust- and risk-models. The model addresses issues related to the trust management life-cycle [30]. These issues are in particularly the trust formatting and evaluation and exploitation. Also autonomous trust decision based on gathered evidence are mate possible. This without prior or little knowledge of the operating environment.

Trust Values

In the SECURE project the format of the trust values is chosen to be inter-operable, privacy enhancing and scalable [52]. The value based on observation, recommendation and reputation. Obviously all this is based on the available evidence an entity can obtain. This however is hard to come by in a real world scenario [30]. The disadvantage of this approach is that it is not defined on a real world definition of trust. Simply because none actually exists. However, in a digital environment such a model is required to make it even possible to model a trust value at all [52].

The trust value itself may be an arbitrary number, a number plus a piece of evidence or only the piece of evidence without any numerical interpretation done. If one of the last 2 is chosen then it is obvious that the larger the store of piece of evidence is the more accurate the trust value (cn) become. However, there will arise issues on scalability and storage [1] [52].

Then there is another choice to be made, which of the three option above is the more useful. Certainly for recommendations a value-only approach is not sufficient enough. This is because the evidence used for this calculation may be counted double, may overlap with other recommendation values and therefore pieces of evidence may be counted multiple times [52].

[1] Obviously if you want to store all pieces of evidence you can gather of all entities you encounter, the storage required would be massive. The same obviously holds true of the scalability of the surrounding system and its data retrieval and trust value calculation capabilities

Providing just the pieces of evidence without the trust value will not help either in this case. The hole notion of recommendation is build around the fact that the recommending entity its trust value is of interest. That the pieces of evidence are of interest as well has only to do with the above mention reasons.

The project also defines a triple layered trust information structure where the bottom layer holds a list of pieces of evidence, the middle layer holds a structure with two types of trust value and the top layer holds a combined trust value calculated form the values out o f the middle layer. The values in the middle layer are based on observation and recommendation [52]. With this structure it is possible for entities to request all or only parts of the trust information structure. How evidence it gathered is discussed in section 3.4.4.

3.4.2 Using Trust in Uncertain Environments

In the previous sections we have seen some definitions related to trust in the SECURE project. In section 2.4.1 we have given a definition of the human notion of trust for so far this is possible. In this section we will look at other aspects of the SECURE project that have to do with risk, building trust and the SECURE software framework.

Trust is inherently linked to risk [7]. If we do a step back and consider not just SECURE but look a this in a broader perspective, it can easily be shown that this indeed is a correct assumption as is explained in section 2.4.3. The risk factor in any trust-mediated action is decomposed by its possible outcomes. The outcome is dependent on the other principals trustworthiness and the intrinsic cost of the particular outcome. Although risk factors can span multiple values.

The SECURE request analyser uses this inexorable link to combine evaluations from the risk evaluator and the trust calculator to decide if an action should be taken or not. The trust calculator does not only provide information directly to the request analyser. IT also passes information to the risk evaluator which uses it to select the correct cost-pdf (probability density function). A good example of how this works is given in [7] in section ”Risk analysis”.

Let us turn our attention to what trust really consists of. It is fundamentally based on trust information [7] which is encompassing:

- Evidence from observations
- Recommendations from partially trusted 3rd parties

These two sources of information combined allow to form a dynamic opinion of trust about a principal.

All observations are evaluated against the principals expected behaviour which produces experience. This range of experience values reflect the effect of observed outcome in relation to the expected outcome [7] [1]. The outcome of all this is classified in two sets: trust-positive and trust-negative. All evidence from observations is aggregated with past evidence to obtain an interaction history (reputation).

The other part of the dynamic opinion, recommendations, is mostly useful when observations are imprecise or lagging. Observations have more influence than recommendations. An initial trust-value must be established to allow low-risk interaction. In these cases there will be no or very little evidence through observation available in which case the value has to be build

[1] Note that the expected outcome essentially is your reference which is used to compare to your observations or perception. It are just different terms for the same comparison made as is done in PCT

up out of recommendations by partially trusted 3rd parties [7]. Multiple recommendations are possible, they are weight according to the trustworthiness of the source.

Delegation is possible in SECURE. However the differences between delegation and recommendation are that delegation is only allowed to similar entities that are considered experts, whilst recommendations are gathered from any entity in the environment [7].

It is all fine and well to be able to reason about trust, risk, trustworthiness and making access decision with this information. But without a feasible implementation that works in a heterogeneous system, it is quite useless [7]. Therefore a software architecture framework including trust management algorithms is designed for SECURE.

When a principal is requesting interaction, the request analyser will request information from three sources: entity recognition component, trust calculator and risk evaluator. The ER component is responsible to recognise previously seen and new entities and is discussed in the next section. It can be called by any other component to obtain entity recognition capabilities. Further, more specific, details on the usage of identities and the gathering of evidence will be given in the next two sections.

3.4.3 The Role of Identity

In this section we will look at the role of identity in the trust decision. Whilst it is more based on entities and not on real-world identities for privacy reasons, most of the technical and related trust concerns are similar to those faced in a user-centric environment. To cover identity recognition and trust management, the SECURE project uses an advanced TSF (trust_risk-based security framework).

The TSF system is designed out of three components [52]:

Decision component: This component is called when another entity this entity is called by another entity, it decides what action should be taken, it has two subcomponents:

- Trust-engine based on evidence from observation or recommendation
- Risk-engine evaluates risk invoked, it chooses the right action to keep an appropriate cost/risk balance

Evidence component: As the name suggest it gathers evidence which is used by the trust- and risk-engine in the decision component (the managed life cycle). the evidence consists of recommendations and the difference between the expected outcome of a certain action and the real outcome [1].

ER component: The Entity Recognition module (ER) needs to identify and recognise the entities connecting to the system

In the remainder of this section we will look more closely at the ER-component.

In this system entities are seen as pseudonyms to the real-world identity. A problem with this approach is that it is impossible to check if a real-world person has created one or more pseudonyms or not. Site-effect of this is that when an entity is recommended by another entity it is impossible to check if the recommended entity is not owned by the same person that gets recommended. With this trick a person could hack the system by increasing the

[1] This exactly is what the reference and perceptual signals from PCT are. Note that the difference of these two is the error signal (see the previous chapter)

recommendation or trust in a certain entity till above a certain threshold. PKI systems however are no solution to this problem as they have shown to be unable to provide a link between these pseudonyms and a real-world identity [52] [1].

The terms ”entity” and ”identity” are used mixed in the research for this project. This is because the identity used to gain access is a virtual identity which is not mapped t a real-world identity by the system [52]. The reason for this is that an identity by itself says nothing about the behaviour of the person behind it. Therefore ”who is” is less important than ”can I trust”. The identity by itself does not imply privilege, the trust value does. This also holds for the Jericho architecture where identifying the requesting party is a bit more important as it is the basis of most policies. It still however will not imply privilege and therefore is only the first step and not the complete process, if it was it would not need trust management.

The entity recognition process is used also for authentication. This is achieved by binding an external visible identity to the recognised entity. This is achieved by recognition the other entity by its observable attributes. The recognition process has four steps [52]:

1. The system triggers the recognition mechanism
2. Some detective work needs to be done to recognise the entity using the available scheme(s)
3. Retention of information relevant to the recognition process
4. Actions taken upon the outcome of the recognition process, this may include a confidence value in the recognition result

Because of the data retention sept, even if it is discriminative, there is an inherent conflict between trust and privacy, both depending on data in the opposite direction [2]. The only way however that trust relationships can be build is that there is no per-transaction usage of identities. The reason is simple, if for every transaction an entity uses a different identity the system could never find enough evidence to base a trust decision on. Whilst this is highly privacy enriching, it would knock the basis out under any trust management system that is using evidence, and therefore reputation, as foundation for its trust decision.

3.4.4 Evidence Based Trust Broker

In this section we will look at the evidence gathering side of the story and also will discuss the design of the trust broker model which is used by the SECURE project. This trust broker functions in part quite the same to the one defined for the Jericho architecture. Although, as we will see, it does not broker in trust, it brokers in evidence and trust values whilst leaving the final calculation to the requesting entity.

The first issue we will come across is the fact that if you want to calculate an accurate trust-value, where and how much evidence would you require. As said before, the more the better but that has its scalability issues. The other problem in the respect to where to get it is related in the fact that the evidence is spread across multiple domains. Entities may work around this

[1] Public Key Infrastructures will not store any real-world identity information, so it is near impossible through these systems to find out who in the offline world a certain key-pear actually belongs to. Key-signing parties do use real-world identity documents to authenticate before a trust-exchange is agreed upon, but this information will get lost in the translation to the actual digital key signing process in the PKI (GPG in this case) system

[2] For trust management the more data is present as basis for a trust decision the more accurate the decision can be. For privacy the less data that needs to be exposed to the outside world the higher the privacy protection is. These two values are at odds and will ultimately lead to brokerage in privacy to gain trustworthiness

problem by querying old and already visited networks for recommendations. This is based on the fact that entities there may keep evidence about an entity for a certain amount of time after it disconnected. However, some restricting or even preventing measurements may effect such connections. These may include such things as technical limitations in devices i.e. storage limitations for evidence, technical impediments in the network itself i.e. firewalls, network transition, range or bandwidth limitations and privacy policies can prevent connecting from the outside to issue requests for evidence [30].

However, besides all this some options for gathering evidence exist, these are:

- Ask entities for possible recommenders for evidence
- Ask neighbours (more easily accessible) for possible recommendations
- Broadcast a request for recommenders
- Ask brokers to suggest good recommenders

There are some issues with these options though. When using options one, the integrity and therefore the correctness of the recommenders can not be ensured, the second option can not guarantee the quality of the data and the third option is very bandwidth intensive and should only be used as a last resort [30].

Before we are discussing the trust broker model itself, some security requirements have to be defined as well. The two most important ones are privacy and integrity of data. No further definition is given, it is only stated that if these options are well defined in the system, secure gathering of evidence is possible whilst also ensuring the privacy of the entities involved [30].

The SECURE project also defines a triple-value (s,i,c) that can be used for the representation of evidence, that is the historical record of interaction. In this triplet s denotes the value of positive interactions whilst c denotes the negative ones. The i-value is used for the level of uncertainty [30].

Trust Broker Model

For the trust broker model [30] gives the following definition:

> The Trust Broker Model provides an infrastructure to allow trust information to be accessed across different system boundaries in a secure manner.

As already stated above, this does mean that the trust broker itself does not take any trust decisions itself. It leaves this responsibility to the calling entities. In this respect the trust broker defined here is fundamentally different than the one defined in the Jericho architecture.

First we will define the preliminary situation, that is the definition of domains and entity classification. The system considers the entire Internet as one global domain. As this is an absolute unfeasible approach for trust management, local domains are defined by technical and/or location criteria. Besides this there are too many diverse types of infrastructure which makes one unique approach unfeasible. And as already said, the scale of information about entities, storage and retrieval, are absolutely unfeasible on a global scale [30].

All identification between entities will happen within one local domain. All other ”local” domains are seen as remote domains. The system also assumes that all domains are interconnected and the no isolated isle do exists. Entities are classified by two characteristics: can they

communicate with other domains, can the perform basic trust broker functionality. If an entity qualifies for both points it is a so-called strong entity, if not it is considered to be weak.

Now that these definitions are out of the way, let us define the tasks of the trust broker itself. The trust broker is responsible to track all information related to all interactions going on between entities inside a single domain. To accomplish this task all requests for trust values are done through the trust broker, accidentally also hiding the identity of other entity which does increase privacy, and because all entities within a single domain are trusted by all entities [30].

A request is handled as follows:

- A queries the trust broker about a certain transaction with B
- The trust broker now queries his repository for similar transactions with B
- If found it requests evidence from the found entities for this transaction about B
- The gathered evidence is passed to the calling party A

In the system no single entity will always be THE trust broker. The current trust broker and the potential trust brokers in the domain are known as the trust broker pool.

Within this pool of potential trust brokers a re-election process may take place if a certain number of requests for it are received by various entities which may happen on the failure of the active broker or on its disconnection from the local domain. If a re-election is started, it will use the following criteria in its process [30]:

- The trustworthiness of the potential trust brokers
- The policies of the entities themselves, does it want to become broker or not
- The entity its abilities
- The entity its mobility.

only strong entities are eligible for the trust broker pool. The capabilities may have to do with storage and bandwidth availability. The mobility of the entities is a measurement on how likely it is the entity will disconnect from the domain i.e. laptops are more likely to move away than desktop systems. The repository mentioned is kept up-to-date between the entities via the trust broker protocols and an extra backup-service in the domain. These protocols are defined in [30] and will not be explained here in detail.

This also holds for the procedures in place for secure evidence gathering. The following five procedures are defined:

- Registration of a new entity to a local domain
- Registration of an existing entity to a newly elected trust broker
- The creation of signed evidence
- Evidence gathering within the local domain
- Evidence gathering from remote domains

As specially the 5th procedure is very hard to do.

As already stated, all entities must trust the elected broker and the entities that can become broker do need to be a bit more reliable than ordinary entities. All trust brokers also have an

embedded digital signed certificate issued by a certifying agent. This last bit is assumed in all trust broker protocols which were mentioned above.

We conclude this description of the evaluation process included in the trust broker. As all recommendations are transferred through the trust broker, it is able to use multiple recommendation to derive: combined results, attraction of the trust value and recommendations representing a local domain. Future considerations lay in the areas of feasibility and performance.

3.4.5 Summary

As we have seen in this section, the SECURE project tries to use a human notion of trust as basis for access decisions. With this goes the inexorable problem that trust as it is is very had to define properly in the social sciences let alone in a digital environment. Part of the solution is not only to look at trust but at the associated risk to transactions as well. Trust in the mean time is build up out of recommendations by other entities and observations done by the entity itself. These two sources of information combined form a dynamic opinion about another entity. In fact, the bigger idea behind the whole system is not really trust as such but one of the main, and definable, building blocks of trust: reputation. This is clear by the fact that all trust calculations in SECURE are based on evidence gathered mostly out of observations and where absent or not precise enough via recommendation as well. Most of the issues faced in this project, specially on the definitions of trust, are also faced in this paper. Therefore, it is a fair assumption that the design in chapter 5 will be evidence, reputation, based as well. Although we are using a bit of a different approach by basing the system and the reputation definitions on the PCT, in essence the problem is similar.

3.5 Summary

In this section we will look back a the tree sections of research overviews in this chapter and summaries which elements are useful for the requirements that are defined in the next chapter. By now we have seen that most research that is said to be in trust management is not or at least using a very limited definition of trust. Most of this is trust in credentials and therefore trust in the certification principles used. These certification principles nearly all of the time are based on PKI (public key infrastructure) and therefore asymmetrical encryption technologies.

Some systems like PolicyMaker and KeyNote 3.1.2 attempt to use a compliance checker as means of digital trust compliance. However both of the systems lag any definition of trust at all and simply check credentials and assertions against written policies which makes the not much better than your average access control list. Granted it is a dynamic version of an ACL, but that is where it stops.

In 3.2 we have come across some interesting models of trust management which are not directly credentials based but still use the technique as a security mechanism. However, the system is more an example of what would nicely be a triple-a framework dealing with authentication, authorisation and accountability. The only spark of trust found here is in the delegation of access-rights which seem to rely on non-digital and offline hierarchies within the organisation than on a digital trust definition.

In 3.2.1 in this same section we look at something which is closely related to trust. We proposed in the previous chapter that trust itself actually is buildup, in part, from reputation evidence. In this section we have seen some quite interesting arguments for using and how to use a reputation system in a digital context. This certainly will be interesting for the requirements in the next chapter.

Section 3.3 takes a very different design aspect as it comes to trust management systems because it does not deal with your average computer network architecture. However we do show in 3.3.1 that in the end pervasive computing and the Jericho architecture are not that much different than they would look at a first glance.

In the remainder of this section e look at some approaches to trust management and the use of identities and identification in this field of research of which the approach to a user-centric identity management is of particular interest. One special project within this field of research is handled separately.

The SECURE project discussed in section 3.4 has as one of its design goals to use a human notion of trust as their trust definition. In their design they use the parts of the real-world trust definition that have to do with evidence and context [1].

If you look at SECURE through the definition of PCT, you will notice that certain similarities emerge. The SECURE project talks about observation and expected behaviour. It does not take too much imagination to see that an observation of a certain event is similar as the perception of this same event. Likewise, expected behaviour and reference information are similar. With this information it can be argued that SECURE does use some elements of PCT in its design. Which is not that surprising as it uses the human notion of trust as core design issue.

There is however one major difference, the outcome of the above mentioned evaluations is used as experience not as disturbance or error signal for which a correcting action needs to be defined. Because of this subtle but important difference, a one-on-one mapping is not possible. Fact remains that SECURE is a valuable source of information for the design in this paper.

[1] Evidence and context are the only parts of any trust definition that are measurable and therefore usable in a digital environment, the two elements combined could be defined as reputation which is an essential element of trust

Chapter 4

Requirements for a Trust Management System

In the previous chapter we have given an overview of research done which claims to be in trust management. We have found that most solution leave a lot to be desired and are nowhere near the level required for a truly de-perimeterised open network architecture. The only project that comes very close in this respect is the SECURE project discussed in section 3.4.

In this chapter we will define the requirements for designing a trust management system for a fully de-perimeterised open network architecture. We will do this by first defining the requirements for identity verification, followed by the definition and requirements for digital trust decisions and finally the requirements for the entire system. Two of these sections, digital identity and digital trust, also have a subsection discussing the relevant Jericho view on these respective topics.

4.1 Digital Identity

In this section we will look at the digital identity and identification requirements that are part of our system. Verifying that the other party is who he or she claims to be is a first step in establishing trust or at least that the person is not lying about his or her identity. If the actions of that person are trustworthy or not is besides the point in this section.

Identity is a complex concept to define in the day-to-day life as in the online world. The concept of identity comprises the characteristics that make a person remain the same for every offline or online transaction, but in itself an identity has to be unique for each individual. As specified by [reference] a digital identity refers to the online representation of a user's identity, and the identity of those entities (machines, digital services, other users) with whom the user is in interaction [55]. A digital identity used in an electronic/mobile transaction can be the digital representation of a set of claims made by one entity about itself or another entity. Another definition for digital identity is given by [2]. So, digital identity is a computer representation of an active entity that can be physical (such as a human, a host system, or a network device) or can be a programming agent. However, an identity in a digital transaction needs to reflect the identity of real-life entities such as human beings.

Authentication is establishing an identity by checking that indeed the identity in use corresponds to the entity that it claims to be and thus is said to be authentic. The goal of authentication is to permit only to authorised users to access to different network resources. In a digital world, the entity claiming to have a particular identity asserts its claim by providing verifiable

information to the authenticating entity [2].

Further, we will look at the user-centric model. We follow this by making a distinction between a real world identity and a digital entity. We continue with a short discussion of the Jericho position paper on Federated Identities. We end with a set of requirements for the identity verification part of the trust management system.

4.1.1 Real World Versus Digital Identity

In section 3.4.3 we already discussed some of the trade-offs between real-world identities and digital identities. We see that using pseudonyms as digital identity might enhancing privacy, but also make it near impossible to link back to a real-world identity which is required for identity verification within Jericho. In this section we discuss another paper dealing with this trade-off.

A couple of important statements are made i.e. user control, of personal information, in the realm of digital identity promotes a culture of accountability and trust and acknowledgement and/or recognition make it know that a person does exists [3]. For digital identities it holds that recognition comes from information pointing to a person.

One of the basics of authentication is the fact that a person is consciously aware of the fact that he or she has the same identity and can claim this identity in an IDM-system. In fact, IDM-systems use this effect for authentication purposes when the person can proof he or she can pass a certain set of authentication challenges [3] [1]. If the identity can be displaced by a profile, the sense of being a person, personhood, dissolves. Authentication may fail for some unknown reason. Because the person has no control over the information in the profile, he or she does not know why and can possible not find out. Allowing the person the control over their own identity information is a solution to this issue.

It seems that writers have been aware of the fact that if a person is not in control over his own identity, this causes problems. Since the time of the Greeks, stories ranging from Odysseus in the cave of the Cyclops, Twelfth Night to Dr. Jekyll and Mr. Hyde illustrate this point all too clearly [3]. At least two practical problems can Arise which will be discussed below.

The first issue is that a person's body may change whilst the person himself does not. This makes recognition difficult and its why a disguise works sow ell. It is also the reason why id-card photo's are re-taken at regular periods. The very practical part in IMD is defined by the use of biometrics. Faces, fingerprints and other body parts change over time or may be lost altogether. This has grave implications for the accuracy or even the feasibility of biometrics as authentication method. Biometric researchers are aware of this fact and therefore use two criteria, stability and distinctiveness, to select appropriate traits for biometric usage. Where stability is trait that does change slowly and distinctiveness is a low change that two or more people share this particular pattern. As no biometric trait is entirely stable, biometrics can only be used to some level and degree of identity verification [3].

The second problem is the exact opposite, the body stays the same yet the person changes. A nice demonstration of this is given in some cases of legal exemptions if insanity or a similar fact can be proven which would make the accused unaccountable for his actions. This also gives troubles with identification i.e. people forget passwords. If a system recovers from this forgetfulness, it may ask for physical means of identification i.e. id-cards or birth certificates. However, disasters like Asian tsunami and Hurricane Katrina show that even that is no infallible mechanism of identification [3].

[1] This is also one of the three basis elements of authentication being: what you are, what you know and what you have

The paper also lists a set of identity properties. We will lists these below without explaining them in detail. Identity verification may be a first step in any trust relationship, it is not the main topic of this paper. The properties of identity as defined in [3] are as follows:

- Identity is social
- Identity is subjective
- Identity is valuable
- Identity is referential
- Identity is composite
- Identity is consequential
- Identity is dynamic
- Identity is contextual
- Identity is equivocal

It is clear that none of the tree basics of identity verification can survive on its on to fulfil the task set [1]. We propose, as we will do in the requirements in this section, that at least two or even all three elements must be used for identity verification. This, as specially, in high security data access.

4.1.2 Federeded Identity: the Jericho Position

Many user authentication schemes today still use the deprecated model of usernames and passwords. Specially with large numbers of user-id's to manage, it is no longer a feasible option. Therefore proposals have been done for federated identities where users can use one set of credentials to authenticate with multiple companies that have agreed to work together [27]. The Federated Identity model has been suggested for business-to-business communication where one company manages the credentials and system authorisation for systems run by the other company. Problem with most federated models however is that one company may have to give up control over key assets to another company, which is in a privilege position, to allow issuance and validation of user credentials. An additional issue arises when credentials are coupled with personal information. As specially when this is passed to other companies and/or country borders. This would create privacy issues because of the disclosure of the personal information to other parties without the permission of the user. Most of the systems limit to authentication of human users, Jericho itself however needs to be able to authenticate devices, application and resources as well [27].

For a really open and perimeterless Internet, stronger and simpler authentication methods are required that also meet the business requirements for different and changing degrees of trust (Jericho commandment 8, appendix D). All privacy issues must be visible met as well to allow for equal partnerships.

Many Federated schemes do not match the business requirements. Users who are issued a set of authentication credentials within specific a domain, must be able to use them in another

[1] The three elements are: what you know i.e. passwords, pin-code, answers to specific questions, what you have i.e. id-cards, save-word tokens, smart-cards, certain hardware, what you are i.e. biometrics, the sound of your voice, the way you walk or other measurable behaviour

domain. The other security domain must be able to check these credentials with the original issuer. Therefore the issuer in both domains must be able to supply authorisation information which can be combined to form the final authorisation permission [27]. No privileged identity provider is required and peer-to-peer authentication needs to be supported. Also no homogeneous credential formatting should be required, systems need to be flexible enough to support multiple formats. Different credentials can be used for different situation, e.g. role-based. This may also effect the trust level that is used.

Making a clear distinction between credentials and attributes will clarify data usage, privacy protection and will ease the introduction of new authentication technologies that in the end could replace the username/password schemes. Passing shared-secret credentials to other companies should be discouraged as the risk of compromise is simply too great. Instead seamless pass-through authentication or identity assertions should be used. This holds true for information passed between organisation as well as for information passed between component layers (Jericho commandment 8, appendix D). And finally, individuals should be able to chose which attributes are used for a given transaction e.g. home or work contact details, credit card info.

The position paper [27] ends by stating seven challenges to the industry which we will not list here. They summarise what is said above.

4.1.3 Identity Requirements

For the identity verification process we will propose a hierarchical system that can handle multiple methods of identification. These methods are be divided in three categories:

- What you know
- What you have
- What you are

We propose that at least two methods from different categories are required for a full identification, or at least an attempt at identification verification. Below we will shortly describe each category and its possible methods. Please note that this will by no means be exhaustive listing as new methods may be developed or others may better fit certain implementation criteria.

What You Know

In this category we have the age old favourite of nearly all current identification systems, the password or passphrase. I has been prove to work if strong enough and not written down in a piece of paper or similar. Another often used example is the pin-code. In general everything that can be memorised by the human users will count in this category. This directly is also the absolute weakness of this form of identification because we either forget the information we need to remember or the information is too difficult to remember in the first place and we result in writing it down to be able to use it. This also is precisely the reason why one method will not be enough.

What You Have

Here we can think about smart cards, safe-word tokens and other cards, random readers. However, a card or token can be stolen and used by someone else without a problem. This

is the reason that most of them are protected by at least a numerical pass code to restrict the usage by unauthorised people. The combination illustrated above directly gives an example of a combination of identification methods. Only having the card or its pin-code will not help you in requesting identification verification.

Another method may be a certificate that is provided to the user by a trust third part, like the government, that can serve as a digital passport, obviously protected by a passphrase. This certificate needs have the property that it is unique and completely and directly verifiable, either by the certification authority or a trusted external party that is able to decrypt and verify the contents of the certificate. If it, additionally, is possible to create authoritative sub-certificates holding less person information so that the exposure of person information can be adapted to the level that is required for certain identification criteria and access levels, you have a system that is verifiable and privacy enhancing at the same time [1].

Other methods to satisfy identification criteria are not directly apparent and could only serve as additional methods if we are not sure enough that the person indeed is who he claims to be. In this respect you can think about GSM polling to a phone with a known number and SIM-card, checking of hardware-addressing on the local network [2], additional certificates based on a public key infrastructure (PKI). A last possible resolve may be found in using one-time-pads, either pre-issued or issued via i.e. text messages, to establish a certain level of verification.

What You Are

Now we enter the realm of biometrics which we already discussed in section 4.1.1. As we stated there, biometrics is a tricky subject in this respect as some features changeover time, are easily forgeable like finger-printers or a simply not unique enough leaving the chance that there are more people sharing the same characteristic. Another very big issue arises when your biometric identity gets stolen, i.e. fingerprints. It is impossible to ask and issue a new set of fingerprints. If this happens, that method of identity verification is immediately rendered useless.

If we take all this in consideration, we propose that biometric identity verification is only used if absolutely necessary i.e. for the highest security levels. Secondly, we propose that at least two methods are used or where this is unfeasible a method is chose that is the least susceptible to forgery, the slowest in any changes to it over time and the most unique feature available. We admit however that this might well be the proverbial needle in a haystack.

Conclusion

Digital identity verification is not and never will be 100% foolproof. However we can try to come as close as possible to verifying that a certain person is who eh or she claims to be. For this reason it is required to have multiple levels of identification preferably from at least two of the above discussed categories. Where a combination of the first two is preferable because of the properties of biometrics.

The last part that is missing is an extra level in the hierarchy that combines the information form the three separate control systems, we require one control system for each category. This upper level is also responsible for setting the required methods per category and will combine the values of certainty, obtained from the verification process, to one verification result that

[1] An open standards based system that fulfil these criteria is the Higgens Project `http://www.eclipse.org/higgins/`

[2] Although hardware-addressing is easily spoofable and therefore no means to authenticate by or even identify with

is passed up to the decision levels. With this multi-level design, it is easy to replace or add new methods to one of the three categories without requiring extensive changes to the entire identity verification system.

4.2 Digital Trust

As we have seen in section 2.4.4, it is nearly impossible to come up with a definition for real-world trust that is complete, non-descriptive and objective all at the same time. However, with the lessons learned we can define a system that uses some of the elements to come up with a workable digital trust definition. We will use this definition to draw up a set of requirements for our trust management system.

4.2.1 Modelling Digital Trust with Perceptual Control Theory

Even by using PCT we have been unable to define a clear cut model of trust in a real world scenario (see section 2.4.4). We did however find that one of the main constructs of trust is reputation and that reputation in turn is based on all available evidence about a person either negative or positive.

By trying to define digital trust as close to human-based trust as possible, we have very similar problems as were faced by the people of the SECURE project [7]. Like them we are forced to compromise to a reputation-like evidence based approach. As trust itself is undefinable, even with a behaviour explaining theory like PCT, this is the only possibility.

So we are left with the following definitions:

Trust: An undefinable value made up by personal experience, feeling and a person's reputation

Reputation: A combination of evidence obtained from personal observations and third party recommendations

Recommendation: An evidence based opinion by a third party

Observation: Any evidence of actions and behaviour gathered through direct contact and transactions

Evidence: Specific detailed and objective information about a person's behaviour without any value attached

All these definitions are based on the same aspect as they all deal with evidence and/or the evaluation of evidence or already evaluated evidence.

We define digital trust as: an evaluated set of evidence gathered through observation and/or requested through recommendation. What we actually have defined, is a system based on reputation. As said before (section 2.4.5) reputation is one of the basic elements of trust. As such, it is also the only building block that can be measured in at all. Although the exact mathematical model behind this will not be part of this paper and is an option for future research. See also sections 2.4.2 and 3.2.1 which handle reputation and reputation systems respectively.

The definition can be broken up into two parts: observation and recommendation. We need to separate these for our definition in PCT. The reason for this is simple, for recommendation the evidence is already evaluated, we can re-evaluate it if we want to at any time, whilst for observation this is not the case. We need to define a reference- and perceptual signal for both

cases [1]. Note that we do not define a controlled variable. As the controlled variable is the input, it is similar to the perceptual signal [2]. We do however define the error signals, or more precisely, the possible actions resulting from the possible error signals.

In the case of recommendation, it is actually quite simple. The perceptual signal is the trust value that we obtained, the reference signal is the trust value we would really like to see. If these do not match, we have two possible error signals, a negative and a positive signal. It is obvious that if the trust value is higher than what we want it to be, the only course of action is to send it ”up one level” and let the system correct it [3].

On the other hand, if the signal is negative, we have a problem. Two possibilities arise now: re-evaluate the evidence obtained along with the trust value or let the system correct the reference signal to a lower value. The result of the first option may yield a solution if the way the recommender evaluated the evidence differs from the way it is done in the local system, if not we will need to referred to option two after all. The result of the second option will be that the overall trust value is lower than we want it, can we correct this error signal directly? The question is more, do we bother at this stage to correct it. Perceptual Control Theory, and as specially the Method Of levels, teaches us not to. Therefore the error remains and is send upwards in the hierarchy where a higher, or even top-level, control loop needs to deal with it.

The other scenario, that of observation, is a more complex one as we do not deal with already evaluated evidence but with observed actions that need to be evaluated and possibly stored. The first thing to do to make this even possible is, define what we understand as an observation. Only with that definition we are able to define the different signals and possible actions.

What we see as evidence of a transaction is difficult to define completely. If we take the definition too stringent we limit the capabilities of the final system, yet if we make it too loosely we will be overwhelmed with pieces of evidence. In general it is true that the more evidence you have, the better and more fine-grained your reputation and trust values will be. But as we have said before, in section 3.4.4, this has scalability issues. We will define evidence as: a transaction, or part of a transaction. With this somewhat general definition, we can define the evaluation of pieces of evidence as: the result of the transaction measured by the level of success or failure.

Now that we have a measurement, we have a definition of the perceptual signal because we observe the transaction. The reference value is defined by the accepted success factor of a transaction. This factor however is calculated, and set as reference signal, at least one level higher and is comprised of: the known or initial trust value of the other party, the value of the transaction and the risk that is taken by accepting this transaction.

Obviously, as opposed to the recommendation scenario, there is only one possible action if an error signal occurs which is either positive or negative. This action is to send the error signal ”up one level” and let the system deal with it. The reason for this is simply because the higher levels in the system have a broader perception on the environment. It may even be that the recommendations obtained are positive as well as possible observations from earlier transactions. It would not be much of a reputation if we discounted that information.

What we actually have defined in this system are the low level control loops that process evidence. Higher level control loops will need to take care of combining all information to a fully

[1] Note that the re-evaluation process is similar to the observation process in respect that we run the evidence obtained from recommendation through the observation system.

[2] Observe that we are talking about a very simple PCT-system with, at this level, no parallel inputs and therefore a very simple perception

[3] The only correction possible will probably be that the reference signal will be raised to match the required value

calculated trust value. We will not define the exact mathematics for this calculation, but leave it open for future research.

4.2.2 Circles of Trust

The circle of trust we discuss in this section is a framework designed by the Liberty Alliance [1] which is a global organisation that ”provides a holistic approach to identity”. They put trust in a centralised position within any transaction on the web. The question they raise therefore is, who you can trust with what.

The Liberty Alliance states that support for the law is lacking in digital contracts and signatures. For this reason a contractual framework is required which is necessary for a legally binding circle of trust. This includes agreed upon obligations, rules, and remedies that will govern such relationships [53]. They claim that this is very useful in the perspective or risk management [2].

The contractual framework deals, amongst others, with the topics like: roles, rights and obligations, confidentiality, enforcement and remedies and entrance and exit of members. The Liberty Alliance states that implementing such a framework will lead to more robust and trustworthy relationships [53].

We will not discuss this framework here in detail. The fact of the matter is that the framework as proposed in [53] is more a contract and a legal framework than an open network architecture. I certainly is necessary to have a set of rules and to make sure we can enforce any legislative rules and legal action upon breaking of these rules, it however has nothing to do with trust at all.

Only the collaborative model has a some resemblance to the Jericho architecture as well as to the ”prisoner’s Dilemma” problem which we discussed in section 2.4.4 and appendix B. The other two models are either too restrictive by not allowing or even expecting members joining or leaving, consortium model, or because control lies with one single part, centralised model, which makes it a single-point-of-failure which is an aspect absolutely useless in an open network environment.

The general idea of a legally binding agreement may yield some resolve. The problem however remains the same, how do you treat unknown and/or new identities. Another potential issue here is the fact that such a CoT will certainly span across natural borders of jurisdiction. So long as international legislation has no resolve for cross-boarder online trade and the enforcement of agreements, having a CoT is very nice but might simply proof to be a bag of hot air when someone breaks the rules in the end.

Finally, the question posed ”who do we trust and with what” is not even answered. The only measurement that is taken are the rules and obligations and the promise of the CoT members that they will abide by them. In theory this sounds quite nice, but if it stands up to the barriers of international law remains to be seen.

4.2.3 Trust and Cooperation: the Jericho Position

It is a well known fact that any e-Commerce transaction relies on a certain level of mutual trust. This is trust in believe that the other party will fulfil his or her part of the transaction as promised. However, trust in the business sense relies primarily on contracts and the enforcement of the

[1] http://www.projectliberty.org

[2] As we have seen in section 2.4.3, trust and risk are each other’s opposites

promises made in them [24]. For this principle to work in an e-Commerce environment, a registration process is required which can register and verify the identity of each involved party in a transaction. However, such processes are hard to automate and therefore expansive which resists the growth of e-Commerce.

Current federated identity systems aim towards federating identities between different parties of a single supply chain or multiple once. However, there is a need for federation between organisations. These need to make the federation process more easily and also allow for new technologies like reputation for sharing trust information or a legal framework for standardised contract templates [24].

Trust is an essential element in any electronic collaboration. However, processes like trust management and registration are expansive and often complex due to differences in policy requirements. De-perimeterisation requires the ability to share reputation information between organisation to reduce cost [24]. The position paper defines trust as a vital pre-condition for successful collaboration. Central to this definition of trust lies the contract which essentially is made up out of a set of rules specifying what each party should do and accountability mechanism to handle failures by any party. Note that a contract, as specified here, does not have to be a legal contract, it could simply be a code of behaviour as specified within a certain community [24].

Trust also is a precondition for cooperation as it allows two parts of a transaction to take part separately, usually separate in time. An example of this is a payment and delivery cycle of a product. The buyer pays because he trusts that the product indeed will be delivered as promised. The concepts of cooperation that are specified in [24], have also been discussed, in another context, in section 2.4.4 and [49]. The question is posed how one party decides to trust another party, based on the proposed contract and the perception of the other party his past performance. Good performance in similar area's makes it more likely that the other party will be trusted. A record of performance, either good or bad, constitutes reputation [24]. The question is raised how two strangers may ever come to trust each other. Two mechanisms are proposed for that: parties may share reputation information with parties they trust or a party may decide to trust a stranger in a small way initially.

The remainder of this position paper [24] defines a trust architecture and extends this to include the concept of de-perimeterisation by using a trust broker. This is, however, not relevant in the context of this section and will therefore not be discussed.

4.2.4 Trust Requirements

With the definition of digital trust in section 4.2.1, we have already made a start with defining the trust requirements for the system. In that definition we defined a hierarchical model consisting of two parallel control systems for observations and recommendations respectively. Apart form any mathematics that are needed to do the calculations, the only thing we need to have is a new level in the hierarchy on top of the two parallel control system. This level is required to combine the trust values obtained from recommendation and observation to one trust value for the given identity. With this level we complete the trust evaluation.

4.3 Trust Management System (TMS)

In this section we combine the requirements from the previous sections to come to a complete system. We will first discuss the hierarchical nature of the system. We follow this by defining the levels of decision which are based upon the requirements defined in the previous two sections.

We end with the requirements for the inner workings of the trust management system (TMS) itself.

4.3.1 Hierarchy of Control

It is obvious from the requirements we defined in the previous sections, that we are dealing with a multi-level control system. Although it depends on the part of the system how many levels are present, at least the system in its entirety will be multi-level in its design. We follow closely the model of HPCT as we discussed previously in this paper in section 2.2.

The reason this hierarchical model is chosen is a direct consequence of the reason to chose PCT for our digital trust definition. As consequence of that choice, it is reasonable to build the remainder of the system using the same definitions. Therefore all levels, from the lowest observational level to the highest decision level, is seen as one or multiple control loops where some will work in parallel and on possibly different tasks, whilst others are on different levels of the hierarchy and are connected via the error signals coming up and the action/behavioural output, defined as reference signals, coming down the hierarchy.

In general the hierarchy can be divided in a lower component level and an upper decision level. The reason for this is the simple fact that we propose a clear segregation of duties on the identity verification and trust calculate parts. Although we admit that some information is required by both systems, most information is not.

Another reason to propose this design model is that it will be very easy to add new components to the system as well whilst keeping the decision structure more or less unchanged. It may be required to add a risk analysis component to evaluate the risk involved in any given transaction. If we chose to add such a component, it is clear that this risk analysis must have no effect on the trust value being calculate and vice versa. This is because we think that having an independent, and therefore possibly objective, risk analysis done which is separated form the evidence available on the participants in the transaction, may benefit the final decision.

The reason for this is quite simple. We have two forms of risk to deal with, the risk of the transaction as such] and the risk of untrustworthy behaviour by one of the participants in the transaction. We want to see these two as separate instances. It is left up to the decision level system to combine them. Note that this a different approach than was taken by the SECURE project where the risk-component does make use of the calculate trust value to, in part, base its risk analysis on [7] and section 3.4.

4.3.2 Levels of Decision

Ultimately, there is only one decision to be taken: access granted or access denied. However, to come to this decision not only do we have to consider the information passed to us from the lower component levels, we also need to decide what initial levels to send down as reference signals and if these are not met, what to do about the resulting error signals coming back up. Even here we propose a segregation of duties in the decision-levels but only for the initial decision. This means that for every component present in the lower layer, there is at least one decision-loop controlling the component. For our system this means that we have at least two loops, for identity verification and trust value, working in parallel. As consequence of this we also will have at least one loop that is on level two of the decision system. This loop combines the final values coming from the component control loops.

In all cases there will at least be three levels in the system. The two described above and one top-level decision loop. This final loop is the fail-over in the system. If the decision loops below

it can not resolve the error signals they have been passed, this one will have to deal with them. This also means that if this loop fails, the system as a whole fails. How this is accomplished is left to the design and possibly even left as an implementation specific choice.

If so required, levels may have multiple loops dealing with different aspects of the decision they have to make. We have used this multi-level option in the trust value component for the recommendation evaluation defined in section 4.2.1.

4.3.3 System Requirements

With the definitions given in this section and the two previous section, we end up with a nearly completed set of requirements. The system needs to be build on a two layer design, one for evaluation and one for decision control systems. We require that the interface between these two layers is clearly defined and, if at all possible, will not be changeable. The reason for this last requirement is that we can easily replace, rewrite or even extend and of the two parts of the system without having to extensively modify the other part as well.

Unfortunately, we expect that a complete segregation of duties and knowledge of the different levels in the hierarchy will not be possible. As specially in the top of the decision hierarchy, we need to be able to deal with any unresolved error signals originating from any level in the system, even the lowest once in the evaluation control systems. However, if such an evaluation control system is well designed, those errors will be resolved, or at least combined with enough information, for the decision level to be able to make sense of the occurring problem. This property however is a main feature of the hierarchical model of PCT and therefore one of the reasons we chose it to base the requirements of the system on.

For every level in the hierarchy we require the following complete definitions to be completed:

- Perceptual signal, what does the control system require as input
- Reference signal, what does the control system expect as its internal reference as set by the next level
- Error signal, what can go wrong with the comparison in the control system
- Action/behaviour, what can be the results of any error signals that can occur in the control system

With these requirements clearly defined, it should be relatively easy to implement the system in a multi-set of ”black boxes”, each having their defined input- and output signals and a list of expected actions.

As final requirement, each level needs to have a timing constraint or at least a definition on how long it is expected to need to complete its task. As most complete decisions will certainly traverse multiple levels in the hierarchy and as higher levels will take more time to complete their task, we need to take care that the system will not become too slow. With the speed of modern hardware this should not be an immediate issue, but it is something to keep in mind when designing a system where every level is in some way or another depending on the levels below and above it.

Chapter 5

Conclusions

It is interesting to see that a firm belief in being able to find a definition for trust is something, you have to admit later, which is impossible. This research project started out from a technical background and with a technical scope to define a trust management system. Now, at the end of it, we find ourselves without a trust definition (see sections 2.4.4 and 2.4.5) and with a lot of related work done in the non-technical field of social science.

Whilst we have not been able to define trust in a usable way, we did find out that reputation as such is a building block for trust. Even so, it is the only measurable building block we could find. Unfortunately, this is exactly the opposite result of what we were trying to accomplish. On the other hand, it is not directly the end of it either.

As we discovered, a project from the related research field in Pervasive Computing (see section 3.3), faced similar problems in their struggle to define a trust management system on the basis of human trust relationship (see 3.4). Their solution, to use an evidence based reputation- and risk-analysis system, is similar to what we propose to be the result of the requirements in the previous chapter. Experience as evidence, observation, is also a combination made in Perceptual Control Theory [42] which is useful in our hierarchical requirement definitions as well.

The main goal of this research project was to design a system in which the human factor, in the way it made its decisions, played a prominent role. During the project however we found out that by using the Perceptual Control Theory (PCT), we could also apply this design goal to the entire system. The background of this theory in electrical engineering and the modelling principles the theory uses, which are similar to those used engineering, helped greatly in adapting its principles for a technical design.

It is not the first time this has been attempted. Another project [46] also used PCT as the basis for their design. It is a choice in our requirements we found to be very useful. Because of the hierarchical structure of PCT, it was quite easy to form the requirements into a layered system. As the hierarchy also allows for parallelism in the system, it became possible to define a clear segregation of responsibilities between the various levels in the system.

In section 1.5 we posed the question: ”How can we define a trust management system within the Jericho architecture?” With four additional sub-questions. It is unfortunate that, because of the struggle to find a definition of trust, we were not able to answer them all. We did define digital trust with the help of PCT (see sections 2.4.4 and 4.2.1), or at least something usable in the requirements for a trust management system. We also defined the requirements fro digital identity evaluation (section 4.1.3), digital trust decisions (section 4.2.4) and the trust management system with a hierarchical decision construction and multiple decision levels (section 4.3.3).

However, designing the trust management system and defining its security and privacy issues and pitfalls are the questions that will remain unanswered for now. However, in the end it may simply be the trade-off between privacy and data access that is the main question. At least in a user-centric identity system, you are in control of what part of your privacy you are willing to trade.

5.1 Future Research

In a research setting, as was used for this paper, there will always be things that were not done, questions not answered or new questions raised. Either because of scope or because of time constraints. In this section we will list a couple of these issues important enough to warrant attention.

One thing that comes to mind is the extension of the human-based trust management system to include devices and applications. As the behaviour of such digital entities are mostly predictable, is simply using their advertised behaviour as only guideline enough or are there other factors that play a role in this. This as well goes for their need for access to data, how do you decide that a given entity has access rights and how do you detect it is not compromised. Having certificates stored on the device is clearly not an option and simply authenticating on a hardware- or MAC-address, which is easily spoofable in most cases, is a non-solution either. So we pose the question: how to extend a human-based digital trust management system to include non-human devices and applications in a trusting and secure way?

A second issue, which is closely related to the previous one, is how does the trust management system access its own data or data it receives through recommendation. As Jericho proposes that all data items should secure themselves and that a trust broker should be consulted to request access, where does that leave the trust broker itself. Obviously, the trust broker can not simply go the a higher level trust broker as this would not solve the issue at all. We do not want to store the data the broker, and therefore our trust management system, needs unprotected because where is the sense in protecting data in the first place as this vital link in the chain is not protecting it [1]? It is related to the first question posed because we are talking about applications. We list this separately though because it is a special kind of application and a vital part of the Jericho architecture. We pose the question: how can we secure data that is needed by the trust management system in such a way that failure of the system does not institute an immediate data breach? And secondly: if we do secure this data, where should the trust management system ask permission to access it, who should grant it and on what basis?

A third point, which we already mentioned at least twice (sections 2.4.2 and 4.2.1), is that of the mathematical backing of the trust management system. If we could have designed a system that does not need such basic mathematics behind it, we would have given it a fair try. However as it is, even this system needs some numbers eventually. In the end all evaluated evidence needs some value of trustworthiness which can be combined to other pieces of evidence to form the reputation value and therefore the value somebody is trusted by in the system. This may well be combined with a score of certainty on the identity of the user and how sure we are about its correctness. We therefore pose the question: what mathematical concepts and functions are required to come up with a useful and clear trust value? Please note that: the mathematics must support the system and absolutely not the other way round. A starting point

[1] Note that the data we are discussing here is comprised of pieces of trust evidence and may included very sensitive privacy information as well, which certainly holds for the identity data that is stored and processed in the system

in the literature can be found in [51].

Another issue that has to be dealt with is the practical matter of what we think is useful as evidence and what not. Similarly, we need to decide on how much evidence is required, how long it is stored, specially for observational evidence, In some cases it might be useful to keep all evidence about a certain identity, in others it is probably best to delete everything after a certain amount of time has elapse. But doe we delete it all or do we keep a summary for future reference. There is a lot to be decided on this issue and these are nearly all related to procedures and regulations. So we pose the question: what do we constitute as evidence, how do rate it, use it, store it and lose it if so required?

Related to this last question has undoubtedly everything to do with rules and regulations and is the final decision that needs to be made before we can fully implement and integrate the trust management system into a production environment. A coherent set of security levels needs to be defined including the level of trustworthiness that is required to be allowed access. This is s separate discussion on which roles are allowed access and who can adopt them in the first place. Obviously, if you do not have the right role to access a piece of data, your trustworthiness value does not matter a bit. However, which roles there are and who can adopt them is a question that is of no concern to the trust management system. Only the results are of interest, because if we know who you are and we know you haven't the right role to access this piece of data, what is the point in even calculating your trustworthiness to see if you are allowed to access it? So we pose the question: What level of identity certainty and trustworthiness is required for which level of data clearance? Provided the user has the right role to access it, if such restrictions apply.

For clarity reasons we repeat the questions posed in this section below:

- How to extend a human-based digital trust management system to include non-human devices and applications in a trusting and secure way?
- How can we secure data that is needed by the trust management system in such a way that failure of the system does not institute an immediate data breach?
  - If we do secure this data, where should the trust management system ask permission to access it, who should grant it and on what basis?
- What mathematical concepts and functions are required to come up with a useful and clear trust value?
- What do we constitute as evidence, how do rate it, use it, store it and lose it if so required?
- What level of identity certainty and trustworthiness is required for which level of data clearance? Provided the user has the right role to access it, if such restrictions apply.

5.2 Acknowledgements

Credit where Credit is due, I begin by thanking prof. dr. S.D. Swierstra and drs. P. van Oostrum for their assistance and support in realising this graduation project. Over the years I have enjoyed the various contacts we had on subject matters and now also for my graduation research. It has been a privilege working with you. I would further like to take the opportunity to thank Marco Plas for his belief in my abilities and his time and effort in making it possible for me to make my goal of graduating with a doctoral degree a reality. I believe that his belief, enthusiasm and persistence, gave me the strength and possibilities to finish what I started so many years ago.

Another big factor have been the people involved with the Perceptual Control Theory. Because of their effort and time I have been able to use this theory in the context of this research. Special thanks go out here to Dag Forssell, Bill Powers, Frans Plooij and Marco Plas who did initially point me in the direction of PCT after all. However, extra gratitude is due for Dag Forssell for his limitless inspiration and his continual support in providing readable versions of the various articles and books that were used for this paper. I also want to thank Dag for the use of his figures which are prominently present in this paper as well. It has been an honour and privilege working with you and I enjoyed our many conversations by mail, phone and in person. Last but not least I want to point the reader to his website [1] where information on all used books and various online articles can be found. Special attention is due on his book "Management and leadership" [15] and the freely available e-book "PCT - A Book of Readings" [23] where various articles and book extract can be found as well [2].

Special thanks are also due to Alina Stan who was my p-manager and coordinator during my internship at Capgemini and helped by providing various literature documents, overall comments and structuring, Ellen Blijenberg, Paula Blaauw, Natalja Jonkers and Deborah Chedi (F55 secretariat) for there continued moral support and assistance, Kas Clark for his assistance in layout, grammar and spell checking of this paper, Richard van Monsjou for designing the front page and arranging the printing process, Wim Duijvestein for his moral support and friendly advice and last but certainly not least I want to thank my family and friends for believing in me and there continued moral support and assistance were possible.

[1] http://www.livingcontrolsystems.com

[2] Download details can be found here: http://www.livingcontrolsystems.com/sampler/readings.html

Appendix A

Bill Powers on PCT and Trust I

From: Bill Powers
Date: Tue, 01 Jul 2008 15:17:43 -0600
Subject: Re: pct and trust

Hello, Andor –

I'm not familiar with the field of "trust management." It's not clear to me whether this refers to a person trying to justify trusting someone else, or to a person trying to get someone else, or some institution like a bank, to trust him. In the first case, it seems to me that the problem is how to decide what evidence is sufficient, while in the second case the problem is how to convince someone else that you are trustworthy (whether you are or not). I suppose that's the same as figuring out what evidence the other person would consider sufficient. But do we then provide the actual evidence, or select evidence that we think will impress the other, while we keep counterexamples well hidden? And if the other person appears trustworthy, which is he doing?

In either case, I think the basic problem is whether or not to trust one's own perceptions as being true representations of reality. There is often a conflict between what we want reality to be and what we can prove is true. The proving part involves testing, looking for alternate ways of checking what is really Out There. We can't ever look directly at reality without relying on our own perceptual systems, so the proofs always have to be indirect and statistical, as well as relative: we judge some perceptions in terms of other perceptions.

Ideally, we want to trust what is trustworthy and distrust what is not. The mistakes we can make are the remaining two cases: distrusting what is actually trustworthy, and trusting what is not trustworthy. As you say in the paper, this seems to involve past experience, and I would add that it relies on the methods we use to analyse past experience.

I tend to judge other people not so much statistically as on the basis of my understanding of how they are organised. If I can find out what their reference conditions are, I can see whether they appear to be acting against their own wishes, in which case I would tend not to trust their promises. That sort of judgement isn't based just on what they have done in the past; it's more model-based, the question being whether the promised behaviour is consistent with what I understand the other person to want. Does it make sense that this poor Kenyan lottery winner wishes to give me half of his winnings?

Sometimes I have decided to trust someone as a way of encouraging that person to be more trustworthy. But at the same time, I have already discounted the loss that will occur if this strategy doesn't work. I once gave a beggar in a MacDonald's restaurant $10 to buy himself breakfast, and asked him to bring me the change. He actually did bring close to $5 back to me, and when he gave it to me he asked why I thought I would get any money back. I just said

”You brought it back, didn’t you?” and put the money in my pocket. So that time it worked. But I didn’t give him any more money than I was prepared to lose. So did I trust him? Or was I exhibiting trust just as a means to an end?

Does trusting people give them the experience of being trusted, so they can see whether they’d like to repeat that experience? Or does it just show them that you are a trusting person, so if the other person happens to want to take advantage of that, you provide a convenient target? People have varying motives (higher-level reasons) for wanting to be trusted, or to trust others. It’s probably a mistake to generalise and try to find some characteristic that everyone has.

Perhaps what this comes down to is simply an analysis of the ways people find out whether they will benefit by trusting others, or by being trustworthy themselves – or else by doing the opposite. I think we can find examples of all the possibilities.

Best regards,

Bill Powers

Appendix B

Bill Powers on PCT and Trust II

I received this personal E-mail from Bill Powers after explaining the goal of the trust management research done in this paper. This E-mail is a follow-up in the short mail exchange I had with him. The message in the previous appendix precedes the one quoted below.

From: Bill Powers
Date: Wed, 02 Jul 2008 09:42:07 -0600
Subject: Re: pct and trust

Hello, Andor –

I understand better what you're trying to do, now.

I think the first thing you have to ask is whether the problem has a solution. This depends on the degree of security you want. If you want to be absolutely safe in trusting someone on the Internet, or if you want a way to prove beyond doubt to someone else that you can be trusted, I think you know the answer.

Perhaps the only workable strategy is to find a way to equalise the risk, so either person will experience an unacceptable amount of loss if trust is betrayed – and of course a satisfactory gain if it's justified. If a proposal of that sort is rejected, you have a pretty good indication that you should not trust the other party.

I think we have a variant of, or maybe the opposite of, the "prisoner's dilemma" here. In the prisoner's dilemma, if one person testifies against the other and the other remains silent, the first goes free and the other gets a long jail sentence. If neither person betrays the other, both get a short jail sentence. If both betray the other they both receive an intermediate sentence. So betraying the other person will get you either freedom or an intermediate sentence, while not betraying the other will result in either an intermediate sentence or a long sentence. The best strategy, if each person cares only about himself, is to betray the other person.

The prisoner's dilemma is created by a third party who is in a position to make the rules. The prisoners can't simply choose not to be prisoners: they are physically forced to play the game, since the only choices are to betray or not to betray and it's not possible to do neither, or both. They can, of course, try not to play the game. For example, they can testify against the other person in a way that is certain to be proven false at a later time: you could say "His wife told me he came home covered with blood," when in fact he is not married, or when you know that the blood that will certainly be found on his clothes (and yours) is that of a chicken. If you establish yourself as a liar or an incompetent observer, your testimony can't be believed – but you can't be accused of remaining silent, either.

The question of trust, it seems to me, involves asking who makes the rules for this game and how the rules are enforced. If a third party who is empowered to make the rules can be brought

into the relationship (or created!), then it becomes possible to make sure that a person who betrays trust will suffer a loss, but both parties will benefit equally if both remain trustworthy. For example, both parties can be required to post bond (send funds to the third party to cover any possible losses). If the transaction takes place properly, the bonds are returned, minus a modest fee for the service. I'm sure you can think of other ways to handle this.

Here's where PCT comes into it. The "third party" has to be someone or some institution that both parties perceive as being superior. In other words, there has to be some system concept to which both parties subscribe, that is accepted as superior to principles like "me first." A person trying to cheat another person will not agree to risk his own money even temporarily for the simple reason that if he does not live up to the bargain, he will lose an amount of money which is equal to or greater than the amount to be obtained by cheating. The person who lives up to the agreement, or does not violate it, gets the amount promised. One version of this agreement is called, in the US, an "escrow" agreement.

It seems to me that we're now talking about the origins of government and law. If I announce that I am prepared to enter into a contract with anyone else who will subscribe to the same system concept, and if the terms of the contract include giving some tangible collateral to assure performance, I can be sure of not losing anything on the deal, and by putting up my own collateral, I assure the other of not losing, either. But this requires that a third party exist who is considered absolutely trustworthy, or at least significantly more trustworthy than the average participant in an agreement. This party, after all, will be holding the bond money or the collateral, and we have to be sure this party can't just run away with everything. So we have to create this party ourselves and keep it under constant public observation.

Eventually we will figure out that we don't need to put up the whole amount of the transaction as collateral. We can require some small percentage of it to be held by the third party, just as "earnest money," if we also give this third party the physical means to force a defector to make good on any broken promises – to impose a serious cost that the untrustworthy person must pay, including perhaps punitive damages or jail time to deter others from trying the same thing.

But for the most important transactions, there is simply no substitute for requiring 100% bonds, so there is no way to profit by cheating and then disappearing. If you want complete security, I don't see any other way to do it.

So I suggest that the idea of establishing trust must ultimately come to a shared system concept under which only being trustworthy pays off. If you can build that into the Internet, you will have solved your problem. And you will very quickly find out who is not trustworthy: they simply won't agree to play by those rules.

Best regards,

Bill P.

Appendix C

Perceptual Control Theory: Timing of control at various levels

This message was sent by Dag Forssell to the Control Systems Group mailing list (csgnet) in 1992. The year of sending obviously accounts for the included ASCII-art diagrams which I left in for clarity reasons.

However, a critical side note made by Dag Forssell in a recent mail exchange, concerns figure C.1: ”Bill and I agree that my ASCII illustration was well-intended in its aim to provide a sense that higher-level control systems are slower than the lower-level systems, but ends up being confusing and – to the extent you get the impression that control happens in stages – misleading. It is certainly not functional.”

From: Dag Forssell
Date: Thu Oct 22, 1992 6:51 pm PST
Subject: Timing of control at various levels

A lovely sequence today. More eloquent clarification, going up a level and expansion of control over time.

I think time is an important variable here, which deserves recognition and will clarify a number of the concerns we all wrestle with in real time. Time figures prominently in the discussion of AWOL commitment as well. Some thoughts on time:

The hierarchical control mathematics taught by Bill and Rick includes slowing factors and thus time recognition. I prefer to think and teach graphically. In Durango 1991, I presented a chart in three levels, portraying what I called: Timing of control. Two levels are shown here, which is all we need.

This is an attempt to portray that the higher level control system MUST be slower for the combined system to be stable. We demo this easily with hand movements.

I also wrestled with time when I developed and posted on behaviour of perception (Dag Forssell (920926)). The intensity level of control muscle fibre control is here and now. The configuration level position of body part is rather present also. The sequence level driving in progress covers minutes or hours. The systems concept level, I am a professional, covers almost infinite time.

It feels awkward to portray the hierarchy of control without all the levels being focused on the present. In today’s post Bill clarifies that we consider a large range of time in our imagination. This is helpful to me. It occurs to me that the time aspect of HPCT can be portrayed as follows: (In my world, if I cannot graph it, it is not real —:)) The illustration that is referred to here is figure C.2.

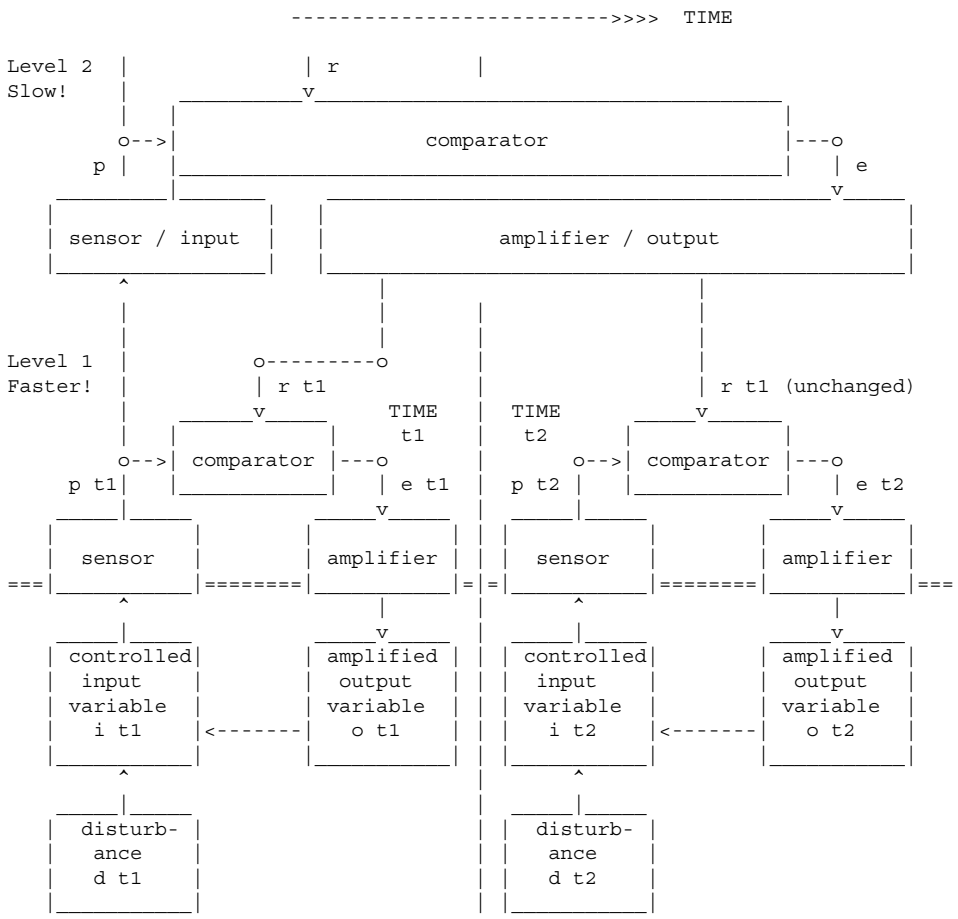

Figure C.1: HPCT, timing of control (two levels)

I have continued to work on behaviour of perception, and will send the graphs to any netter who asks politely with snail mail address.

Best to all, Dag

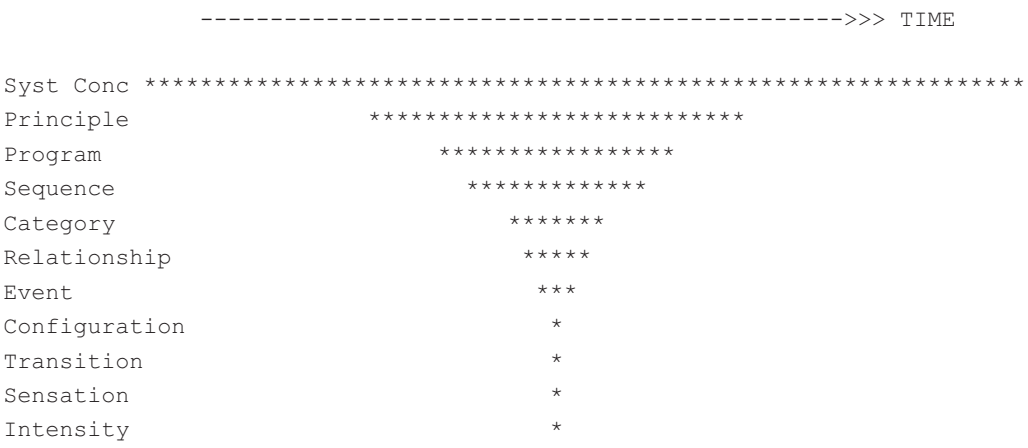

Figure C.2: HPCT, timing aspect

Appendix D

Jericho Commandments

List of Jericho Forum commandments [26] taken from the version 1.2 position paper.

Please note that only the commandments themselves are listed here. All commandments have a list of explanatory items which can be found in the original document [1] on the Jericho Forum website.

1. The scope and level of protection should be specific & appropriate to the asset at risk
2. Security mechanisms must be pervasive, simple, scalable & easy to manage
3. Assume context at your peril
4. Devices and applications must communicate using open, secure protocols
5. All devices must be capable of maintaining their security policy on an untrusted network
6. All people, processes, technology must have declared and transparent levels of trust for any transaction to take place
7. Mutual trust assurance levels must be determinable
8. Authentication, authorisation and accountability must inter operate / exchange outside of your locus / area of control
9. Access to data should be controlled by security attributes of the data itself
10. Data privacy (and security of any asset of sufficiently high value) requires a segregation of duties/privileges
11. By default, data must be appropriately secured when stored, in transit and in use

[1] http://www.opengroup.org/jericho/commandments_v1.2.pdf

Appendix E

Glossary

ACL access control list

AAA-framework authentication, authorisation, accountability or access (sometimes non-repudiation is added as fourth term) [59, 58]

CIA confidentiality, integrity, availability

Control A is said to control B if, for every disturbing influence acting on B, A generates an action that tends strongly to counteract the effect of the disturbing influence on B [41].

CoT Circle of Trust

Firewall system in software or hardware that, given a set of rules, allows or denies the passage of network traffic

GNU Gnu's Not Unix

GPG GNU privacy guard (OpenPGP)

HPCT Hierarchical Perceptual Control Theory

IP-address 32-bit address that is used on the Internet to identify different hosts

PCT Perceptual Control Theory

PGP pretty good privacy

PKI public key infrastructure

SPKI simple public key infrastructure

TMS trust management system

Triple-A see aaa-framework [58]

Bibliography

[1] Evgeny Barannikov. The jericho forum project, authentication and accounting. thesis, Hogeschool Zuyd, June 2007.

[2] Messaoud Benantar. *Access Control Systems, Identity Management and Trust Models*. Springer, 2006.

[3] Bob Blakley, Jeff Broberg, Anthony Nadalin, Dale Olds, Mary Ruddy, Marcelo Thompson Mello Guimares, and Paul Trevithick. At a crossroads: ”personhood” and digital identity in the information society. *STI Working Paper 2007/7*, 2007.

[4] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. *Paper presented at the IEEE Symposium on Security and Privacy, Oakland*, 1996.

[5] Matt Blaze, Joan Feigenbaum, John Ioannidis, and Angelos D. Keromytis. The role of trust management in distributed systems security. *J. Vitec & C. Jensen (Eds.), Secure Internet Programming (Vol. 1603, pp. 185-210)*, 1999.

[6] Thomas W. Bourbon. Three ”dangerous” words. In *Perceptual Control Theory: SCIENCE & APPLICATIONS - A BOOK OF READINGS*. Living Control Systems Publishing, Hayward, CA, Januari 2008.

[7] Vinny Cahill, Elizabeth Gray, Jean-Marc Seigneur, Christian D. Jensen, Yong Chen, Brian Shand, Nathan Dimmock, Andy Twigg, Jean Bacon, Colin English, Waleed Wagealla, Sotirios Terzis, Paddy Nixon, Giovanna di Marzo Serugendo, Ciaran Bryce, Marco Carbone, Karl Krukow, and Mogens Nielsen. Using trust for secure collaboration in uncertain environments. *IEEE Pervasive Computing*, Vol. 2, No. 3, July-September 2003.

[8] Timothy A. Carey. *The Method of Levels – How to do Psychotherapy Without Getting in the Way*. Living Control Systems Publishing, Hayward, CA, 2006. ISBN 0-9740155-4-7.

[9] Timothy A. Carey. About the method of levels. In *Perceptual Control Theory: SCIENCE & APPLICATIONS - A BOOK OF READINGS*. Living Control Systems Publishing, Hayward, CA, 2008. posted to an email group in January 2006 in response to suggestions that MOL be used with couples and in groups.

[10] Brent N. Chun and Andy Bavier. Decentralized trust management and accountability in federated systems. *hicss*, 09:90279a, 2004.

[11] Sadie Creesel, Michael Goldsmith, Bill Roscoe, and Irfan Zakiuddin. Research directions for trust and security in human-centric computing. *Privacy, Security and Trust within the Context of Pervasive Computing*, 2005.

[12] Edward E. Ford. *Freedom From Stress*. Brandt publishing, Scottsdale, AZ, 1989-1993. ISBN 0-9616716-1-0.

[13] Dag C. Forssell. Pct in a nutshell [online]. May 1994 [cited June 2008]. Available from World Wide Web: http://www.livingcontrolsystems.com/intro_papers/pct_nutshell.pdf.

[14] Dag C. Forssell. About this book. In *Management and Leadership: Insight for Effective Practice*. Living Control Systems Publishing, Hayward, CA, 1994-2008. ISBN 0-9740155-5-5.

[15] Dag C. Forssell. *Management and Leadership: Insight for Effective Practice*. Living Control Systems Publishing, Hayward, CA, 1994-2008. ISBN 0-9740155-5-5.

[16] Dag C. Forssell. Perceptual control - a new management insight. In *Management and Leadership: Insight for Effective Practice*. Living Control Systems Publishing, Hayward, CA, 1994-2008. ISBN 0-9740155-5-5.

[17] Dag C. Forssell. Perceptual control - details and comments. In *Management and Leadership: Insight for Effective Practice*. Living Control Systems Publishing, Hayward, CA, 1994-2008. ISBN 0-9740155-5-5.

[18] Dag C. Forssell. Perceptual control - management insight for problem solving. In *Management and Leadership: Insight for Effective Practice*. Living Control Systems Publishing, Hayward, CA, 1994-2008. ISBN 0-9740155-5-5.

[19] Dag C. Forssell. Why study pct? In *Management and Leadership: Insight for Effective Practice*. Living Control Systems Publishing, Hayward, CA, 1994-2008. ISBN 0-9740155-5-5.

[20] Dag C. Forssell. Understanding purposeful behavior [online]. December 1996 [cited Februari 2008]. Available from World Wide Web: http://www.perceptualcontroltheory.org/articles/Intros/1997_Forssell_introguide.html.

[21] Dag C. Forssell. Pct is reverse engineering [online]. 1997 [cited June 2008]. Available from World Wide Web: http://www.livingcontrolsystems.com/intro_papers/reverse_engineering.pdf.

[22] Dag C. Forssell. Once around the loop: An interpretation of basic pct. In *Perceptual Control Theory: SCIENCE & APPLICATIONS - A BOOK OF READINGS*. Living Control Systems Publishing, Hayward, CA, 2008.

[23] Dag C. Forssell, editor. *Perceptual Control Theory: SCIENCE & APPLICATIONS - A BOOK OF READINGS*. Living Control Systems Publishing, Hayward, CA, 2008. A free downloadable e-book is available here: http://www.livingcontrolsystems.com/sampler/readings.html.

[24] Jericho Forum. Position paper, trust and co-operation [online, cited Februari 2008]. Available from World Wide Web: https://www.opengroup.org/jericho/trust_coop_v1.0.pdf. version 1.0.

[25] Jericho Forum. De-perimeterization explained [online]. 2008 [cited Februari 2008]. Available from World Wide Web: http://www.opengroup.org/Jericho/.

[26] Jericho Forum. Jericho forum commandments [online]. 2008 [cited Februari 2008]. Available from World Wide Web: https://www.opengroup.org/jericho/commandments_v1.2.pdf. version 1.2.

[27] Jericho Forum. Position paper, federated identity [online]. 2008 [cited Februari 2008]. Available from World Wide Web: https://www.opengroup.org/jericho/Federated_Identity_v1.0.pdf. version 1.0.

[28] D. Gambetta. Can we trust trust? trust: Making and breaking cooperative relations. *New York, Basil Blackwell*, pages 213–237, 1988.

[29] Mario Hoffmann. User-centric identity management in open mobile environments. *Privacy, Security and Trust within the Context of Pervasive Computing*, 2005.

[30] Dong Huang and Shane Bracher. Towards evidence-based trust brokering. *Security and Privacy for Emerging Areas in Communication Networks*, 2005.

[31] Lalana Kagal, Tim Finin, and Anupam Joshi. Trust-based security in pervasive computing environments. *IEEE Computer*, 2001.

[32] Yiicel Karabulut. Towards a next-generation trust management infrastructure for open computing systems. *Privacy, Security and Trust within the Context of Pervasive Computing*, 2005.

[33] Bruning M.A. Jericho project, trustbroker framework. thesis, Hogeschool Utrecht, October 2007.

[34] Bruning M.A. Jericho project, trustbroking services. thesis, Hogeschool Utrecht, June 2007.

[35] D. McKnight and N. L. Chervany. The meanings of trust. hlisrc 96-04, University of Minnesota, Management Informations Systems Research Center, 1996.

[36] Mohammad S. Obaidat and Noureddine A. Boudriga. *Security of e-Systems and Computer Networks*, chapter 6. Cambridge University Press, 2007.

[37] Marco Plas. Planning for disaster [online]. April 2006 [cited July 2008]. Available from World Wide Web: `http://www.onewalldown.com/jericho/planning_for_disaster`.

[38] Marco Plas. Risky business [online]. April 2008 [cited July 2008]. Available from World Wide Web: `http://www.onewalldown.com/jericho/risk_management_pct`.

[39] William T. Powers. *Behavior: The Control of Perception*. Benchmark Publications, New Caanan, CT, 1973,2005. ISBN 0-9647121-7-2.

[40] William T. Powers. The nature of pct. *Paper presented at the annual meeting of the American Educational Research Association, San Francisco*, April 1995.

[41] William T. Powers. A brief introduction to perceptual control theory [online]. 2003 [cited Februari 2008]. Available from World Wide Web: `http://www.brainstorm-media.com/users/powers_w/whatpct.html`.

[42] William T. Powers. Experience, reality, and hpct. In *Perceptual Control Theory: SCIENCE & APPLICATIONS - A BOOK OF READINGS*. Living Control Systems Publishing, Hayward, CA, 2008. Post to CSGnet September 1994.

[43] William T. Powers. On emotions and pct: A brief overview. In *Perceptual Control Theory: SCIENCE & APPLICATIONS - A BOOK OF READINGS*. Living Control Systems Publishing, Hayward, CA, 2008.

[44] Paul Resnick, Richard Zeckhauser, Eric Friedman, and Ko Kuwabara. Reputation systems. *COMMUNICATIONS OF THE ACM December 2000/Vol. 43, No. 12 45*, 2000.

[45] Richard J. Robertson and William T. Powers. *Introduction to Modern Psychology*. Benchmark Publications, New Caanan, CT, 1990. ISBN 0-9647121-6-4.

[46] Philip Robinson. Architecture and protocol for authorized transient control. *Privacy, Security and Trust within the Context of Pervasive Computing*, 2005.

[47] Philip Robinson, editor. *Privacy, Security and Trust within the Context of Pervasive Computing (overview chapters)*. Springer Science+Business Media, Inc, 2005.

[48] Philip Robinson, Harald vogt, and Waleed Wagealla. Some research challenges in pervasive computing. *Privacy, Security and Trust within the Context of Pervasive Computing*, 2005.

[49] Philip J. Runkel. *People as living things – The Psychology of Perceptual Control*. Living Control Systems Publishing, Hayward, CA, 2003. ISBN 0-9740155-0-4.

[50] Philip J. Runkel. *Casting Nets and Testing Specimens - Two Grand Methods of Psychology*. Living Control Systems Publishing, Hayward, CA, Originally published in 1990, revised in 2007. ISBN 0-9740155-7-1.

[51] Jordi Sabater-Mir and Mario Paolucci. On representation and aggregation social evaluations in computational trust and reputation models. *International Journal of Approximate Reasoning*, 2006.

[52] Jean-Marc Seigneur and Christian Damsgaard Jensen. The role of identity in pervasive computational trust. *Privacy, Security and Trust within the Context of Pervasive Computing*, 2005.

[53] Victoria Sheckler, Piper Cole, Peter Lord, Joseph Alhadeff, Robin Wilton, Jane Winn, and Colin Wallis. Liberty alliance contractual framework outline for circles of trust [online]. 2008 [cited August 2008]. Available from World Wide Web: `http://www.projectliberty.org/liberty/content/download/2962/19808/file/Liberty%20Legal%20Frameworks.pdf`.

[54] Amir Spahibl, Michael Kreutzer, Martin Kahmer, and Sumith Chandratilleke. Pre-authentication using infrared. *Privacy, Security and Trust within the Context of Pervasive Computing*, 2005.

[55] Lara Srivastava, Tim Kelly, Chin Yung Lu, and Lucy Yu. Digital.life itu internet report [online]. 2006 [cited August 2008]. Available from World Wide Web: `http://www.itu.int/osg/spu/publications/digitalife/docs/digital-life-web.pdf`.

[56] Alina Stan. Jericho project, secure communications: 'end-to-end encryption' in jericho networks. thesis, VU University Amsterdam, 2007.

[57] Leon Teheux. The jericho forum project, authorization & endpoint security. thesis, Hogeschool Zuyd, June 2007.

[58] J. Vollbrecht, P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, and D. Spence. Aaa authorization framework. RFC 2904, The Internet Society, August 2000.

[59] J. Vollbrecht, L. Gommans, C. de Laat, D. Spence, and G. Gross. Generic aaa architecture. Technical report.

[60] L. G. Zucker. Institutional theories of organization. *Annual Review of Sociology 13(1)*, pages 443–464, 1987.