Modelling Dynamically Changing Trust-Relationships in Ad-hoc Coalitions

Helge Janicke and Antonio Cau
Software Technology Research Laboratory
De Montfort University
Leicester

September 15, 2011

Abstract

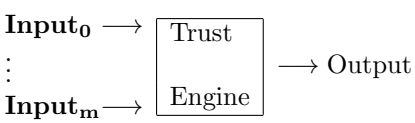
In this paper we present a formal, mathematical approach to the management of trust between agents acting as providers and consumers of services and information. Trust is a relative concept and based on direct experience or reputation as a measure of indirectly reported experience by other agents, but also depends on the context the agents are currently acting, represented by their current environment and/or operational roles. In a military setting establishing and defining trust-relationships between partners in an agile mission group that is part of an ad-hoc coalition force is a difficult and challenging task. A carefully measured balance must be struck as too many constraints impede the effectiveness of the deployed group whereas too loose constraints, or a too free sharing of information, can create security risks. In this paper we propose a policy-based approach to the management of such trust-relationships that combines a crisp approach based on behavioural description and formally defined rules using temporal logic with the notion of trust levels that are defined using fuzzy logic. The key contribution of this paper is that the resulting trust policy model can be formally reasoned about to check whether it satisfies a set of safety properties as well as providing a flexible approach to the expression of policy rules expressing the dynamically changing trust-relationship between various operational units in the context of temporary alliances.

1 Introduction

The changing nature of warfare, where asymmetric operations are becoming increasingly prevalent, poses some unique challenges to the defence community. Network Enabled Capability is designed to assist the military in the rapid reorganisation of resources and dissemination of information, both of which are fundamental to expedient delivery of military capability. However, although commanders are experienced and trained to identify the required resources necessary for an operation, the assessment of capabilities to share information within national contingents and coalition forces is more difficult. Constraints on the free flow of information are set to protect sensitive information and the systems that hold it. Rules, which define the hierarchical flow of information between community members, are set by national security authorities and at coalition level. These result in the creation of multiple domains and systems whose interconnectivity is strictly constrained.

We contend that under certain circumstances this may have a negative impact on the timely delivery of critical information to the individual who needs to act upon it, especially in the context of ad-hoc coalitions. In this paper we explore the use of trust policies to describe and manage the flow of information between different communities of interest. This enables us to investigate alternative scenarios to the traditional hierarchical approach in which the dissemination of information is controlled via policy to provide assurances about the access to information whilst adapting to the dynamically changing relationships prevalent in ad-hoc coalitions.

The key contribution of this paper is a policy-based framework that integrates the ability to define rules based on the behavioural context of coalition partners. Rules determine how trusted a coalition partner is to undertake a certain task and how this trust changes based on the current context and observations of past behaviour. To capture the uncertainty in the specification of such rules we combine Fuzzy Logic with Interval Temporal Logic, yielding a powerful mathematical framework in which rules that determine trust can be defined. The novelty is that not only fuzzy trust decisions can be specified (cf. [15], where trust can be High, Low or Medium) but also trends such as Increasing, Decreasing or Stable that define long lasting trust assessments over a period of time.



The following type of rules for the computation of trust relationships are considered: where $\mathbf{input_i}$ is the ith input and output the output computed by the Trust Engine from $\mathbf{input_i}$ ($0 \leq i \leq m$).

So in order to model a trust engine one needs to (1) characterise the relationship between inputs and output, and (2) characterise the behaviour of the engine in terms of these i/o relationship.

As inputs and output are fuzzy values, i.e., values from fuzzy membership functions (an element from the interval of real values $[0, \ldots, 1]$) the *relation* between inputs and output is usually described by a *collection of trust engine rules*, i.e., one big *Fuzzy Logic formula*.

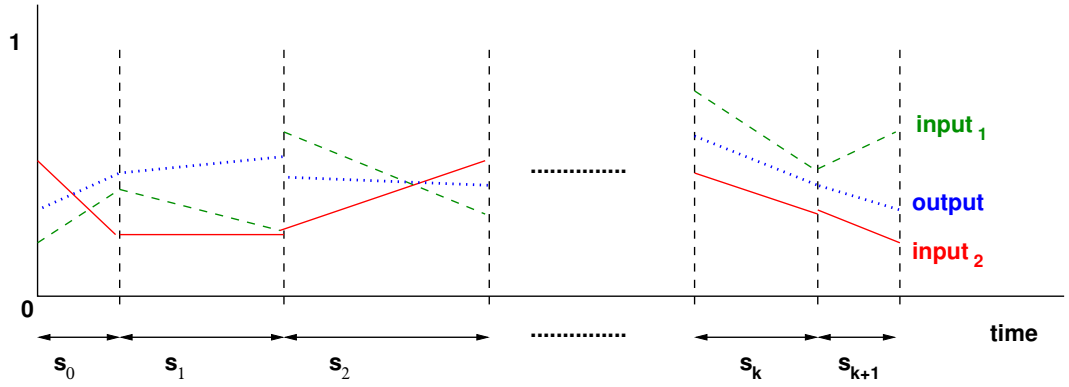


Figure 1: Behaviour of a trust engine

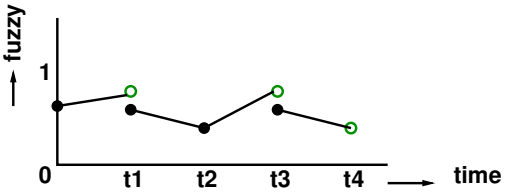


Figure 2: Continuous time behaviour

To model the *behaviour* of a trust engine one first needs to model one *run* of a trust engine, i.e., a sequence of inputs and their corresponding outputs (see Figure 1). The behaviour of a trust engine is then set of all possible runs. However Fuzzy Logic *lacks* constructs to describe runs. By combining Interval Temporal Logic (ITL), a logic for describing the behaviour of systems, with Fuzzy Logic one can describe the behaviour of a trust engine. We will consider continuous time type of behaviours in this paper, i.e. a behaviour is a sequence of continuous phases where in each phase fuzzy variables change according to a spline, as illustrated in Figure 2. Accordingly we will introduce SPITFiRE, a SPline Interval Temporal logic over Fuzzy Relations.

The paper is structured as follows. In Section 2 we review some related work in the area of trust management in ad-hoc coalitions. In Section 3 we present the syntax and informal semantics of SPITFiRE. In Section 4 we show how policy rules that define trust relationships can be defined using SPITFiRE and give examples of such rules. In Section 5 we present a trust architecture for ad-hoc coalitions. We conclude this paper in Section 6, where we also outline our future work in this area.

2 Related Work

The assurance of the availability, confidentiality and integrity of information are predominant factors that decide on the success of military operations [1]. One approach to the specification of high-level protection requirements is [11], where the system is modelled as a composition of information domains and permissible business connections. Whilst providing a good high-level overview of the information system, this approach assumes that the system is static and that business connections and associated security requirements do not change. Especially with the move to network enabled capabilities and the increasing demand for the ad-hoc formation of coalitions this assumption seems too strong [5]. A more flexible way to define protection requirements is a policy-based approach to security-management [14], in which rules govern how the security of information exchanged within such a coalition force can be shared. Grandison and Sloman discuss in [4] how trust influences policy specification and survey a variety of definitions of trust. Jøsang et.al. [6] surveys a series of trust models and categorises them based on criteria such as distribution, discreteness and formalisms used in their specification.

Cahil et.al. [3] provided a formal trust-model as part of the SECURE project that specifies the level of trust based on a set principles and integrates these specifications in trust policy functions that combine trust decisions of individuals. Our approach to the combination of individual trust rules is similar, but defines the combination using Fuzzy Logic operators that are well studied for the integration of values that have a degree of uncertainty. The SECURE framework then combines risk analysis with trust management to yield sensible policy specifications in the light of uncertainty and systematic abuse. Molloy et.al [10] also investigated the relationship between Risk and Trust to overcome the restrictiveness and inflexibility of current access control systems. They take a market-based approach to risk that is based on the notion that transactions in a system represent a risk. They define risk as a finite commodity that can be traded and evaluated their approach against traditional hierarchical security models such as Bell-LaPadula [2]. In another highly decentralised approach Li et.al. [8] particularly address the issue of privacy protection in collaborative environments. They define a state transition model of how trust can evolve based on an agents history of interactions as well as recommendations from other agents that takes the uncertainty of a

collaborative environment into account. With a very similar objective Kosaka and Chatterjee [7] defined a trust model for secure ad-hoc collaborations in which the model defines a trust score between 0 and 100. They use the notion of a Trust Threshold to define when a node in the model is deemed untrustworthy. Fuzzy approaches to Trust modelling, e.g. Machala [9] or Sabater et.al's REGRET [13], take a similar approach by defining a confidence threshold for set-membership. We take a similar approach by defining the threshold as a behaviour of a trust variable over time (See Section 4), with the advantage that this threshold is able to adapt to changes in context and coalitions. More recently Ruohomaa and Kutvonen [12] discussed the notion of (Dis-) Trust in Inter-enterprise Collaborations. Their research is related as it also takes into account the ability of the trust model to adapt to changing business situations, however the do not formally model these changes as part of their model.

3 SPITFiRE

First we briefly recap Fuzzy Logic and will then introduce SPITFiRE, a SPline Interval Temporal logic over Fuzzy RElations.

3.1 Fuzzy Logic

The syntax of Fuzzy Logic (FL) is in Table 1 where x is a linguistic variable, $\mathbb{X}$ a fuzzy set, c is a fuzzy value ($c \in [0,\ldots,1]$), $\mathbf{P}, \mathbf{Q}, \ldots$ are fuzzy propositional variables and $\rightarrow$ is the residual implication and $\otimes$ is the strong conjunction.

fuzzy logic expressions

$$e ::= \quad x \text{ is } \mathbb{X} \mid c \mid \mathbf{P} \mid e_1 \rightarrow e_2 \mid e_1 \otimes e_2$$

derived fuzzy logic operators

$\sim e$	$\widehat{=}$	$e \rightarrow 0$	(residual negation)
$e_1 \sqcap e_2$	$\widehat{=}$	$e_1 \otimes (e_1 \rightarrow e_2)$	(weak conjunction)
$e_1 \sqcup e_2$	$\widehat{=}$	$((e_1 \rightarrow e_2) \rightarrow e_2) \sqcap ((e_2 \rightarrow e_1) \rightarrow e_1)$	(weak disjunction)
$e_1 \leftrightarrow e_2$	$\widehat{=}$	$(e_1 \rightarrow e_2) \sqcap (e_2 \rightarrow e_1)$	(equivalence)

Table 1: Fuzzy logic expressions

Example 1
Given linguistic variables in_1, in_2 and out, and fuzzy sets $\mathbb{H}$ (high), $\mathbb{L}$ (low) and $\mathbb{M}$ (medium). The description if in_1 is high and in_2 is low then out is medium is expressed in fuzzy logic as $(in_1 \text{ is } \mathbb{H}) \otimes (in_2 \text{ is } \mathbb{L}) \rightarrow (out \text{ is } \mathbb{M})$.

Let $\mu_{\mathbb{X}}$ denote the membership function for fuzzy set $\mathbb{X}$ and let $x_1, x_2, \ldots, x_n$ be the elements of $\mathbb{X}$ then $\mu_{\mathbb{X}}(x_1)/x_1 + \mu_{\mathbb{X}}(x_2)/x_2 + \ldots + \mu_{\mathbb{X}}(x_n)/x_n$ denotes the fuzzy set $\mathbb{X}$ with membership function $\mu_{\mathbb{X}}$. For x a linguistic variable and $\mathbb{X}$ a fuzzy set, x is *not included* in $\mathbb{X}$ if $\mu_{\mathbb{X}}(x) = 0$; x is *fully included* in $\mathbb{X}$ if $\mu_{\mathbb{X}}(x) = 1$; and x is a *fuzzy member* in $\mathbb{X}$ if $0 < \mu_{\mathbb{X}}(x) < 1$. Let $\mathbb{X}$ be a fuzzy set and $0 < \alpha \leq 1$, the α-cut of $\mathbb{X}$, denoted $\mathbb{X}_\alpha$ is a (crisp) set defined as $\mathbb{X}_\alpha \widehat{=} \{x \in \mathbb{X} | \mu_{\mathbb{X}}(x) \geq \alpha\}$. Let $\mathbb{X}^\alpha$ be the fuzzy set with membership function $(\lambda x \cdot \text{if } \mu_{\mathbb{X}}(x) \geq \alpha \text{ then } \mu_{\mathbb{X}}(x) \text{ else } 0)$.

Does Fuzzy Logic have operators to assign values to states? From the Table 1 one would expect that $\leftrightarrow$ to be a suitable operator. However, for instance $\mathbf{In} \leftrightarrow 0.2$ will give the *degree* (fuzzy value) of equivalence and is as such **not** a fuzzy state assignment. This is also partially the reason why Fuzzy Logic can not model the behaviour of a fuzzy trust engine. In the following sub-section, we will introduce an operator that can model a fuzzy state assignment and thus the behaviour of a fuzzy trust engine.

3.2 Syntax of SPITFiRE

In order to describe the behaviour as illustrated in Figure 2 we will introduce a behavioural time model that is a sequence of "fat" states, i.e., states have *duration* and within a state, variables change in a *continuous* fashion (see Figure 3).

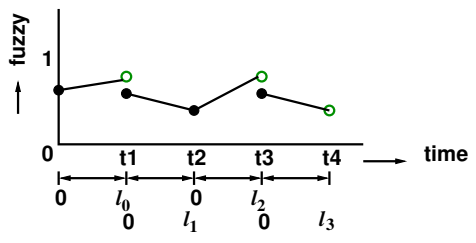


Figure 3: Continuous change within states

An example of a continuous change and commonly used in fuzzy trust engine is a *linear* change. To represent a linear change we need only two points, if the two end points $(0, c_0)$ and (ℓ, c_1) are known then points on the line are characterised by $line((0, c_0), (\ell, c_1))(t) \widehat{=} c_0 + (c_1 - c_0) * t/\ell$ where $c_0, c_1 \in [0,\ldots,1]$ and $\ell \in \mathcal{R}^{>0}$ and $t \in [0,\ldots,l)$. If t changes from 0 to ℓ in a continuous way then $line(t)$ changes from c_0 to c_1 in a continuous way. We take as convention that $(0, c_0)$ is considered to be part of the line and (ℓ, c_1) **not**, i.e., a *left closed* and *right open* line.

Similarly if the change is constant, quadratic or cubic one need respectively one, three and four points. In the following we assume that the possible changes in a fuzzy trust engine can be characterised by a *n-spline* that is defined by $n+1$ points. Let $\text{spline}\langle(x_0, y_0), \ldots, (x_n, y_n)\rangle$ be a n-spline that is defined by $n+1$ control points (x_i, y_i) with $0 \leq x_0 < x_1 \ldots < x_n \leq \ell$.

Fuzzy duration expressions	
$d ::=$	$[r_0 : e_0, r_1 : e_0, \ldots, r_n : e_n\rangle \mid$
	$\mathbf{P} \mid d_1 \overset{h}{\bullet} d_2 \mid d_1 \to d_2 \mid d_1 \otimes d_2$
SPITFiRE formulae	
$f ::=$	$\mathsf{empty} \mid \ell : \mathsf{empty} \mid \mathsf{false} \mid R(d_0, \ldots, d_k) \mid$
	$\neg f \mid f_1 \wedge f_2 \mid f_1 ; f_2 \mid f^*$

Table 2: Syntax of SPITFiRE

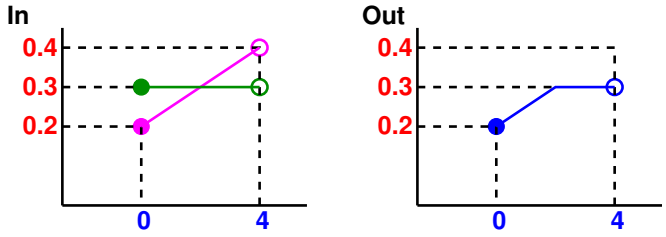
The syntax of SPITFiRE is described in Table 2 where $[r_0 : e_0, \ldots, r_n : e_n\rangle$ denotes a change in the form of a spline defined by $n+1$ control points (knots) $(r_i * l, e_i)$ $(0 \leq i \leq n,\ n \geq 0)$ where $0 \leq r_0 < r_1 \ldots < r_n \leq 1$ and e_i $(0 \leq i \leq n,\ n \geq 0)$ are fuzzy logic expressions; where $d_1 \overset{h}{\bullet} d_2$ denotes the concatenation of two segments described by fuzzy duration expressions d_1 and d_2 segment d_1 takes up h of the duration of the fat state while segment d_2 takes up $1-h$ of the duration; $\mathbf{P}, \mathbf{Q}, \ldots$ denote Fuzzy variables; '$\to$' and '$\otimes$' denote fuzzy operators lifted to duration states; '$(\ell : \mathsf{empty})$' denotes a state with duration ℓ and 'empty' a state with any positive duration; '$(\ell : \mathsf{empty}), \mathsf{empty}$' and '$\mathsf{false}$' are behavioural values; '$\wedge$' and '$\neg$' are the usual Boolean operators; '$R(d_0, \ldots, d_k)$' is a relational operator between fuzzy duration expressions d_j $(0 \leq j \leq k)$; and ';' (chop) and '*' (chopstar) are Temporal (behavioural) operators. An example of a relational operator is '$=$' between two fuzzy duration expressions acting as a fuzzy state assignment operator. We list some derived operators in Table 3.

true	$\widehat{=}$	$\neg\mathsf{false}$	all possible behaviours
$f_1 \vee f_2$	$\widehat{=}$	$\neg(\neg f_1 \wedge \neg f_2)$	or
$f_1 \supset f_2$	$\widehat{=}$	$\neg f_1 \vee f_2$	implication
more	$\widehat{=}$	$\neg\mathsf{empty}$	more than one state
skip	$\widehat{=}$	$\mathsf{more} \wedge \neg(\mathsf{more} ; \mathsf{more})$	exactly two states
$\bigcirc f$	$\widehat{=}$	$\mathsf{skip} ; f$	from the next state onwards
$\Diamond f$	$\widehat{=}$	$\mathsf{true} ; f$	exists a suffix interval
$\Box f$	$\widehat{=}$	$\neg(\Diamond \neg f)$	for all suffix intervals
$\overset{\frown}{\Diamond} f$	$\widehat{=}$	$f ; \mathsf{true}$	exists a prefix interval
$\overset{\frown}{\Box} f$	$\widehat{=}$	$\neg(\overset{\frown}{\Diamond} \neg f)$	for all prefix intervals
$\mathsf{fin}\ f$	$\widehat{=}$	$\Box(\mathsf{empty} \supset f)$	holds in the last state

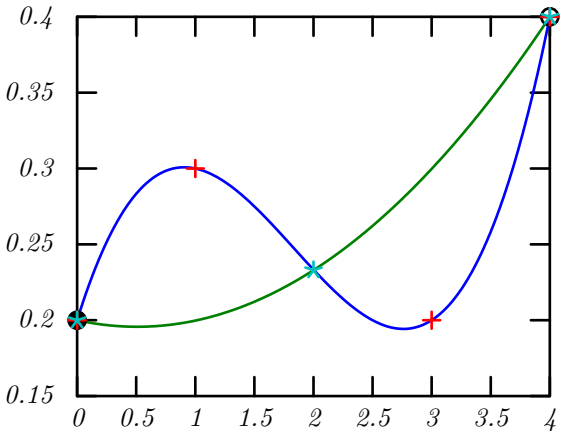
Table 3: Derived SPITFiRE constructs

Example 2 *The following are some example SPITFiRE formulae and their corresponding informal semantics.*

- $4 : \mathsf{empty} \wedge \mathbf{In} = [0 : 0.2, 1 : 0.4\rangle \wedge \mathbf{Out} = \mathbf{In} \otimes [0 : 0.3\rangle$*: one state with duration* 4 *and where* **In** *changes from* 0.2 *to* 0.4 *in a linear fashion and* **Out** *is computed by taking the minimum of the value of* **In** *and* 0.3



- $4 : \mathsf{empty} \wedge \mathbf{In} = [0 : 0.2, 0.25 : 0.3, 0.75 : 0.2, 1 : 0.4\rangle \wedge \mathbf{Out} = [0 : \mathbf{In}, 0.5 : \mathbf{In}, 1 : \mathbf{In}\rangle$*: one state with duration* 4 *and where* **In** *changes from* 0.2 *to* 0.4 *in a cubic fashion and* **Out** *is changed quadratically by using three points from the cubic curve representing the change of* **In***, i.e.,* $(0, 0.2)$, $(0.5 * 4, 0.233)$ *and* $(1 * 4, 0.4)$.



A duration (fat) state describes continuous changes of inputs and outputs of a trust engine over a *finite* time period. A set of sequences of duration states represents the *behaviour* of a trust engine. SPITFiRE, a SPline Interval Temporal logic over Fuzzy RElations, is used to describe the behaviour of a fuzzy trust engine where inputs and outputs change in continuous fashion over finite time periods.

4 Trust Engine rules

We are now able to formalise trust engine rules as follows: Let $pre(input_1, \ldots, input_m)$ be a temporal formula characterising the relationship between $input_1, \ldots, input_k$ over a sequence of states. Let $post(input_1, \ldots, input_m, output)$ be a state formula characterising the relationship between $input_1, \ldots, input_m$ and $output$ within a state. A trust rule $pre(input_1, \ldots, input_m) \rightsquigarrow post(input_1, \ldots, input_m, output)$ is then defined as follows

$$pre(input_1, \ldots, input_m) \rightsquigarrow post(input_1, \ldots, input_m, output) \mathrel{\widehat{=}}$$
$$\boxdot((\mathsf{empty} \vee (pre(input_1, \ldots, input_m)\,;\mathsf{skip})) \supset \mathsf{fin}\ post(input_1, \ldots, input_m, output))$$

If $pre(input_1, \ldots, input_m)$ holds upto the kth state then post holds in the $k+1$th state for all k (see Figure 5).

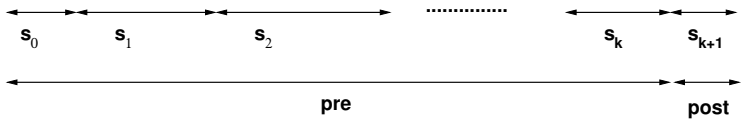
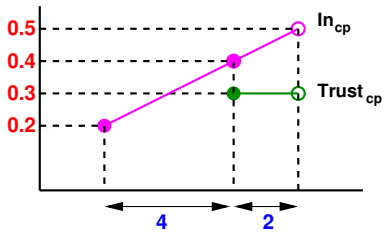


Figure 4: Behaviour of a rule

Example 3



Given the following policy rule: $(\mathsf{empty} \wedge \mathbf{In_{cp}} = [0 : 0.2, 1 : 0.4\rangle) \rightsquigarrow (\mathbf{Trust_{cp}} = \mathbf{In_{cp}} \otimes [0 : 0.3\rangle)$ *representing that whenever the confidence* $\mathbf{In_{cp}}$ *in the information from coalition partner CP linearly increases from* 0.2 *to* 0.4 *in a state then in the next state the trust* $\mathbf{Trust_{cp}}$ *in coalition partner CP is the minimum of* $\mathbf{In_{cp}}$ *and* 0.3.
And given the following input behaviour: $(4 : \mathsf{empty} \wedge \mathbf{In_{cp}} = [0 : 0.2, 1 : 0.4\rangle)\,;\mathsf{skip}\,;(2 : \mathsf{empty} \wedge \mathbf{In_{cp}} = [0 : 0.4, 1 : 0.5\rangle)$ *representing a two state interval where in the first state (of duration* 4*) the confidence* $\mathbf{In_{cp}}$ *in the information from coalition partner CP increases from* 0.2 *to* 0.4 *and in the second state (of duration* 2*)* $\mathbf{In_{cp}}$ *linearly increases from* 0.4 *to* 0.5.
Then the output behaviour is: $(4 : \mathsf{empty})\,;\mathsf{skip}\,;(2 : \mathsf{empty} \wedge \mathbf{Trust_{cp}} = [0 : 0.4, 1 : 0.5\rangle \otimes [0 : 0.3\rangle)$ *representing a two state interval where in the first state (of duration* 4*) the trust* $\mathbf{Trust_{cp}}$ *in coalition partner CP is unspecified and in the second state (of duration* 2*)* $\mathbf{Trust_{cp}}$ *is constantly* 0.3.

In the following section we present the trust architecture that use above trust rules to express the dynamically changing trust-relationship between various operational units in the context of temporary alliances between partners in an agile mission group that is part of an ad-hoc coalition force.

5 Trust Architecture for Ad-hoc Coalitions

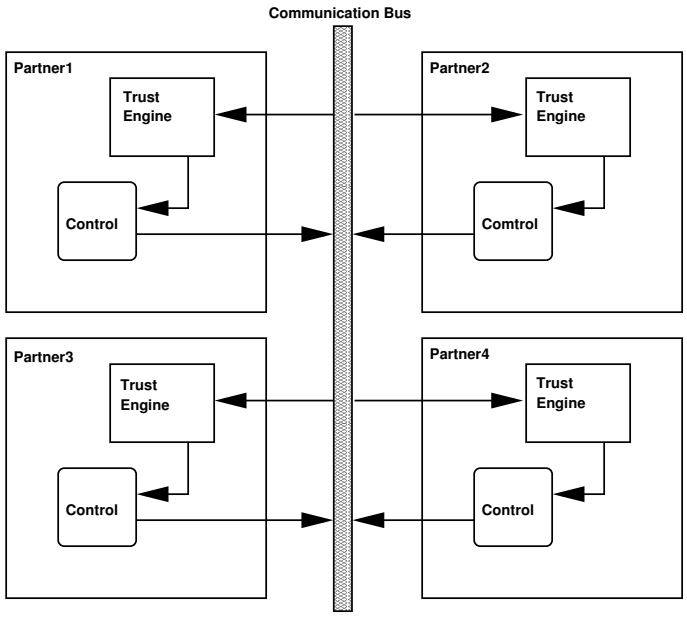


Figure 5: Trust Architecture for Ad-hoc Coalitions

The trust architecture for ad-hoc coalitions consists of a communication bus and coalition partners.

- The communication bus is connected to all coalition partners and information sources helpful for computing trust.
- Each coalition partner has a trust engine and a controller.
  - The trust engine receives input information from the communication bus and sends trust values to the controller. The trust engine has a set of trust rules specified in SPITFiRE. These rules are used to compute trust values.
  - The controller makes decisions based on the trust values computed by the trust engine and also releases information to the other partners based on these trust values.

6 Conclusion and Future Work

In this paper we motivated the need of modelling trust relations between coalition partners on the basis of past behaviour and in a form that easily allows to combine various rules that members of the coalition have. We described how this can be addressed using a combination of Fuzzy Logic to describe vague concepts and uncertain decision about trust with Interval Temporal Logic to formalise how trust relationships evolve over time. The combination of these two mathematical frameworks resulted in the SPline Interval Temporal logic over Fuzzy RElations (SPITFiRE). We presented the logic and its informal semantics and gave examples how a collaboration of trust engines can be specified as a component of a dynamic coalitions decision support infrastructure. The novelty of the approach is that rules can explicitly refer to past observations (or a history of encounters) and specify the affect on trust in the future. This is more powerful than existing approaches, as it a) allows for a declarative description of how trust is build and b) can specify concepts like Increasing, Decreasing or Stable that reflect dynamic change as opposed to just "static" outcomes as "High", "Medium" or "Low".

In our future work we will investigate the integration of SPITFiRE with access control and network management policies to create an integrated system in which the trust model affects the decision making through a managed network.

References

[1] David S. Alberts. *Understanding information age warfare.* CCRP publication series, DoD, US, 2001.

[2] D. Bell and L. Lapadula. Secure computer system unified exposition and multics interpretation. Technical Report MTR-2997, MITRE, Bedford, MA, 1975.

[3] Vinny Cahill, Brian Shand, Elizabeth Gray, Nathan Dimmock, Andy Twigg, Jean Bacon, Colin English, Waleed Wagealla, Sotirios Terzis, Paddy Nixon, Ciaran Bryce, Giovanna di Marzo Serugendo, Jean-Marc Seigneur, Marco Carbone, Karl Krukow, Christian Jensen, Yong Chen, and Mogens Nielsen. Using Trust for Secure Collaboration in Uncertain Environments. *IEEE Pervasive Computing Mobile And Ubiquitous Computing*, 2:52–61, July 2003.

[4] Tyrone Grandison and Morris Sloman. A Survey of Trust in Internet Applications. *IEEE Communications Surveys and Tutorials*, 3(4), September 2000. http://www.comsoc.org/livepubs/surveys/public/2000/dec/index.html.

[5] Helge Janicke and Linda Finch. The Role of Dynamic Security Policy in Military Scenarios. In *Proceedings of the 6th European Conference on Information Warfare and Security (ECIW07)*, pages 121–130, July 2007.

[6] Audun Jøsang, Roslan Ismail, and Colin Boyd. A survey of trust and reputation systems for online service provision. *Decis. Support Syst.*, 43:618–644, March 2007.

[7] Kristie J. Kosaka and Samir Chatterjee. A trust model for secure adhoc collaboration. In Gurpreet Dhillon, editor, *Proceedings of 8th Annual Security Conerenc*, 2009.

[8] Min Li, Hua Wang, and D. Ross. Trust-based access control for privacy protection in collaborative environment. In *e-Business Engineering, 2009. ICEBE '09. IEEE International Conference on*, pages 425 –430, oct. 2009.

[9] D. Machala. Trust metrics, models and protocols for electronic commerce transactions. In *Proceedings of the 18th International Conference on Distributed Computing Systems*, 1998.

[10] Ian Molloy, Pau-Chen Cheng, and Pankaj Rohatgi. Trading in risk: using markets to improve access control. In *Proceedings of the 2008 workshop on New security paradigms*, NSPW '08, pages 107–125, New York, NY, USA, 2008. ACM.

[11] QinetiQ. Domain Based Security. White paper, 2004. `http://www.qinetiq.com/dbsy`.

[12] Sini Ruohomaa and Lea Kutvonen. Trust and distrust in adaptive inter-enterprise collaboration management. *J. Theor. Appl. Electron. Commer. Res.*, 5:118–136, August 2010.

[13] Jordi Sabater and Carles Sierra. Regret: A reputation model for gregarious societies. pages 61–69, 2001.

[14] M. Sloman. Policy driven management for distributed systems. *Journal of Network and Systems Management*, 2:333–360, 1994.

[15] Zhengping Wu and A.C. Weaver. Application of fuzzy logic in federated trust management for pervasive computing. In *Computer Software and Applications Conference, 2006. COMPSAC '06. 30th Annual International*, volume 2, pages 215 –222, sept. 2006.