

Voluntary Economic Cooperation in Policy Based Management

Mark Burgess and Siri Fagernes
Oslo University College, Norway

Abstract

Systems with decentralized authority are sometimes considered to be ‘unmanaged’ or even unmanageable. Promise theory is an approach to policy that assumes complete decentralization of authority. Cooperation between agents or systems is entirely voluntary, so why would agents cooperate in forming policy? By exhibiting the relationship between promise theory and game theory, we propose that there is a natural economic incentive for cooperation in distributed systems with autonomous control. The possibility of trading between agents motivates the definition of a common currency. Our results are especially applicable to the analysis of policy in a Service Oriented Architecture. We derive minimal requirements for the existence of stable Agreements between agents, with or without monetary payment.

I. INTRODUCTION

A promise is a specification of policy asserting that a specified behaviour will fall within particular bounds[1]. Promises cannot be forced onto anyone: they are made voluntarily, and they signal autonomy of decision-making. Promises can be asserted either from an agent to itself or from one agent to another, and each promise implies a constraint on the behaviour of the promising agent. The atomicity and autonomy of promises makes them a tool for detailed analysis of policies, either for finding contradictions and inconsistencies or considering their value to the agents making and receiving them..

The theory of promises is a graph theoretical framework for understanding policy relationships in networks, where bounds and constraints have to be met[2]. In previous work, promises have been assumed predefined. Here, we would like to ask: what is it that makes autonomous agents form promises, keep them or betray them? This is a question that is particularly important when considering agents in an economic framework: e.g. where the outsourcing of tasks is an integral part of a distributed operation, or in a pervasive (ubiquitous) computer setting where agents interoperate on a purely *ad hoc* basis. There are no authoritative controllers arranging and forcing such cooperative promises onto agents.

In this paper we consider the following problem: consider a system of agents that behaves autonomously, in the sense of refs. [1], [2] (this is not to be confused with the weaker idea of autonomy in Multi-Agent Systems[3], [4]). We show that promise graphs can be mapped onto economic games and hence rational economic principles can be used as a tool for predicting the

equilibrium outcome of general policies in agent interactions. Clearly rational game theory is not the be-all and end-all of economic interaction, but it is a stepping stone that will allow us to address more complex decision models in future work, such as reputation and trust, as well as completely irrational choices.

The plan for our paper is as follows. We introduce the idea of voluntary cooperation for agents, and show how such agents can make value judgements about promises. We map this valuation of promises onto a utility function or payoff-matrix for N -agent interactions, and hence find a relationship between games and promises. We posit that the solution of such a game is a natural prediction of a stable policy. We summarize some results that are known from the theory of cooperation, based on two person games and suggest how these results can be interpreted in the framework of computer policy.

II. VOLUNTARY COOPERATION AND SERVICE

Consider an assembly of agents, denoted $n_i, i = 1, \dots, N$, each of which is entirely responsible for its own behavioural policy towards neighbouring agents (i.e. exhibits complete autonomy). No agent takes orders from any other. Examples of such autonomy can be found in software like cfengine[5], [6].

Agents exhibit cooperative behaviour by making promises to a neighbour, if they perceive a value in doing so to themselves. Hence each agent must be capable of evaluating the worth of promises and services made to it by other agents, in relation to its own load of commitments. We assume that each agent has its own algorithm for making this judgement.

So why should an agent choose to cooperate, when there is nothing that forces it to do so, and when there is no guarantee that its peers will do the same in return?

Example 1 (Opportunism): Consider two autonomous agents, agent A and agent B , where A takes the role of a server, and B a client. Servers and clients typically have opposing interests, as the client wants the best possible service for the lowest price available, while the server is more concerned of receiving a high price for as little effort as possible. In an attempt of acting selfishly, A could deliver a service of poor quality, or not at all, while B could be tempted to not pay (enough) for the service.

- 1) What mechanisms are needed in policy for the agents to *start* cooperating (emergence)?

2) What is necessary for the agents to *keep* cooperating, once started (stability)?

These are typically macro-level economic concerns. They concern what is *profitable* for a group as a whole. Yet there is no implication of governance of the assembly of agents from a central point, or committing them to a plan of authority that is decided externally. If we look at each individual agent and assume it selfish, the agents are only concerned with:

- 1) What *benefits* can I gain from interacting with my peers?
- 2) What might I *lose* from interacting with my peers?
- 3) How should I *behave* to maximise my gain, or minimise my loss?

Benefit and loss can be viewed differently by each agent depending on its world view[1], [7]. To predict the overall behaviour of the group of autonomous agents, all these issues need to be addressed.

Services are about committing to behaviour that is valuable to a client. We can use promise theory to define this in terms of interactions between a number of agents. Agents are initially peers that have no predecided roles. It is useful to begin with general definitions that can be specialized to the view of network services later.

Definition 1 (Promise): A promise is an autonomous specification of future behaviour by an agent.

We denote a promise from one agent n_1 to another n_2 by

$$n_1 \xrightarrow{\pi} n_2 \quad (1)$$

where π is called the promise body. e.g.

- $\pi = P$ - is a promise, to another agent A_2 , to perform a service or behaviour P .
- $\pi = U(P)$ - a promise to use/accept or receive the offer of P promised by another agent

A promise body is a tuple, in general, which contains a *type*, (e.g. promise of web service, promise to make a backup etc) and a *constraint* (e.g. web service in less than 5 ms, or backup every hour on the hour). We do not need to define this more carefully here.

Definition 2 (Commitment): We reserve the special term commitment for promises that require an investment of effort on the part of an agent to fulfill, i.e. A commitment is a promise that costs an agent some measurable work or effort.

III. PREVIOUS WORK

In the literature, several attempts have been made to force cooperation of agents in so-called ‘autonomous’ systems. However, the meaning of autonomy is often less strict than ours, e.g. implying only that agents work independently, but can still be commanded.

Mobile ad-hoc networks rely on nodes or agents performing basic services for one another, such as providing routing information and forwarding packets on demand; this is a requirement, as there is typically no central infrastructure for performing these tasks. At the same time, the nodes often suffer from limited power supply, and performing basic, but crucial tasks for the benefit of its neighbor nodes, is very energy-consuming. The obvious selfish choice would be to avoid performing these kind of services, to the certain disadvantage of the other nodes. If a significant portion of the nodes are selfish, communicating with other nodes will be nearly impossible, and the ad hoc network basically non-existent.

A common approach has been to assume certain functionality in the hardware to make it necessary for the nodes to cooperate in order to have the desired functionality. One example is the so-called *nuglet counter* which increases for each time the node forwards a packet on behalf of another node, and decreases when the node sends one of its own packets[8]. Unless the nuglet counter has a value greater than zero, the node is not able to use the network for sending its own packets.

The above is an example of stimulating cooperation between nodes, but this is in no way an example of *voluntary* cooperation. It relies on having some sort of homogeneous tamper-proof hardware for all nodes, which makes it impossible for each node to not cooperate, at least to a minimum, we claim that this is an example of enforced cooperation. It implies an external authority which is obeyed implicitly by all of the agents. In other words, such agents are not have elected to give up their autonomy.

Another approach which is closer in spirit to voluntary cooperation, is the so-called *reputation-based* system. Such systems implement mechanisms for discovering and logging bad behaviour in the agents, which again can be used as aid for other nodes to punish bad behaviour.

Another example is peer-to-peer (P2P) content distribution systems. The main purpose of each system is to distribute some piece of content to as many peers as possible during some period of time. The strength of these systems is their efficiency (it is a form of epidemic spreading), which

is the result of the fact that all peers with a downloaded piece keep uploading to other peers, and hence contribute to further distribution. One important reputation factor, is how long the peers who have finished downloading stay connected after they are finished. There are several different approaches to keep the nodes connected for as long as possible[8].

IV. THE VALUE OF A PROMISE

How shall one measure the value of a promise to the giver or the receiver? A promise offers a *constraint* on a set of values; it has no *defined value* in any form of recognized currency. Moreover, autonomous agents have *a priori* no common concept of universal currency. Each agent is free to decide on some valuation of a promise as it sees fit. It could be a measured level of service quality, or any function of the service quality. A promise by a ‘famous’ or ‘royal’ agent could be worth more than a promise by a commoner. Alternatively it could be a heuristic such as an arbitrary scale of ‘business relationship value’ or other subjective valuation. Such subjective valuations are the basis of economic markets, so we wish to allow these kinds of heuristic judgements.

We assume that each agent n_i possesses a *valuation function* $v_i(\cdot)$ which it can apply to promises given and received in order to determine their value. In general, an agent can only evaluate this function on a promise that it knows about, i.e. one that it makes $v_i(n_i \xrightarrow{\pi} n_j)$ or one that it receives $v_i(n_j \xrightarrow{\pi} n_i)$. The currency of this value v_i is private to the agent n_i concerned.

In some cases, agents will have to agree on a currency standard in order to make agreements. Trading of promises will turn out to be important to stability and reliability—i.e. whether agents choose to keep their promises.

Example 2: The value to Internet Service Providers in Border Gateway Protocol peering promises is known to be based more on fostering good relations and business acumen than on technical measurements[9], [10]. It is important to account for these business intangibles in modelling economic collectives.

We assert then that one should take seriously a multi-currency view of promise interactions. An aim in this paper is hence to motivate these promise valuations, especially those which arise from mutual trade of promises. We begin by exploring the issue of value-driven interactions between the agents, or games.

V. GAMES AND PROMISES

The mechanism of rational, voluntary decision-making used in economics is the theory of games[11], [12]. In this section, we would like to show that there is a natural relationship between game theory and certain constellations of promises which provides a motivation for keeping those particular promises out of all the promises an agent might make. We shall confine our discussion here to normal or strategic form games.

We alert the reader to a potential confusion of notation here. The symbol π is normally used for a promise body, i.e. the details of the constraint being observed in a promise. This is often used also for the payoff or utility matrix in game theory. We shall therefore use the symbol $U_{ijk\dots}$ for the payoff or *utility* matrix of an N player game.

A. Games

We follow Rasmussen in our description of games to avoid repeating well-known definitions[11]. A game is a number of players n_i , where $i = 1 \dots N$, a set of actions, payoffs and information. A player n_i 's *action set* is the entire set of actions or pure strategies available to him. An action combination is a set of one action per player s_i in the game. Player n_i 's payoff or utility is denoted $U_i(s_1, s_2, \dots, s_N)$ and has one of two interpretations:

- It is the utility received at the end of a trial by a player, based on what the player did.
- It is an estimated or expected utility to be considered as a function of the possible or expected strategies a player might choose in a future game.

It is this second interpretation that most closely relates to promises, since we shall identify promises with expectations of actions. However, we should be clear that the latter cannot exist without some measure of the former. This duplicity of interpretation mirrors a similar one in promise theory, in which one can measure the reliability of a promise with a scalar probability value. The probability can then be regarded as a history of a player, or as a prediction of future behaviour.

A *strategy* is a rule that tells player n_i which action s_i to choose at each instant in the game, given his or her information set. An equilibrium $s^* = (s_1, s_2, \dots, s_N)$ is a strategy combination consisting of the 'best' strategies for each player, with respect to the expected payoff. Several such equilibria might exist.

There is a class of games in which the possible actions or moves of a player are the same in each round of a game. Thus each round of an extended game is indistinguishable from multiple games, except perhaps in the information available to the players. Iterated games are special because both *each round* and the entire *iterated game* can be represented as a utility matrix of the same form. Thus we may discuss both extended interactions and one-shot interactions using a utility matrix formulation as long as we restrict attention to such games. We shall do this here.

B. Promises

Promises are static snapshots of policy. A promise graph such as that in fig. 1 can be viewed in two ways: either as an equilibrium over a complete interaction epoch or as a single round in

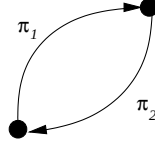


Fig. 1. A simple exchange of promises.

an repeated interaction. To make the move from games with actions to games with promises, we simply replace actions with promises. Any ordering of promises (who makes which promise first) is analogous to an ordering of actions. In a game one considers a strategy to be a choice of actions. Here we shall assume it to be a set of promises made:

$$n_i \xrightarrow{\{\pi\}_i} n_j \quad (2)$$

where $\{\pi\}$ is potentially a bundle of different promises made from one player to another. This bundle might also comprise the empty set.

The payoff or utility received by the i th player or agent in the system is the sum of *its own* private valuations of promises made and received:

$$U_i(\{\pi\}_1, \{\pi\}_2, \dots, \{\pi\}_N) = \sum_{j \neq i} v_i \left(n_j \xrightarrow{\{\pi\}_j} n_i, n_i \xrightarrow{\{\pi\}_i} n_j \right) \quad (3)$$

Any kind of promise can be evaluated, regardless of whether it is a service promise or a use-promise[1] (give or take promise).

VI. CLASSIFICATION OF GAMES

Games are classified in various ways, particularly with regard to cooperation and conflict.

Definition 3 (Cooperation): An interaction is cooperative if players coordinate their actions in order to improve their final payoff (utility). Cooperation might be contingent on available information.

Definition 4 (Conflict): An interaction is a conflict if the actions or intentions of one player can adversely affect the payoff (utility) of another player's actions.

Let us give some examples of these representations of games in terms of promise exchanges.

A. Cooperative, no conflict

Consider two agents helping one another to provide a sufficient level of service for a third party. By helping one another, they are able to work together to meet 3's requirements and secure contracts with the third party and hence both profit: they are not in competition with one another. This scenario is shown in fig. 2.

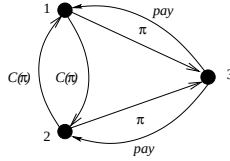


Fig. 2. Cooperative game, no conflict.

Two agents 1 and 2, make coordinated promises to serve node 3. The third party promises to reward both agents for their service, independently. The values of the remuneration promises $v_{1,2}(pay)$ are not related.

B. Cooperative, with conflict

Consider now two agents who bargain with one another directly for services from one another. Both need these services from the other, so the services are valuable (fig 3). By making each other's promises contingent on what they receive from the other, the value each of them receives becomes moderated by the other player. If they cooperate with one another, they can both win a fair amount, but each change can have negative consequences through the feedback. Agent 1

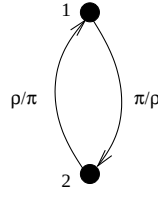


Fig. 3. Cooperative game, with conflict.

promises π if the other promises ρ and vice versa. The values $v_1(\rho/\pi)$ and $v_2(\pi/\rho)$ are inter-dependent.

C. Non-cooperative, no conflict

Two agents providing a service in a free market promise the same service to a third party, without any cooperation (fig. 4). The third party makes its choice and accepts the promised service partly from one and partly from the other. Each is promised a payment. Again, the value of the payment promises are quite independent, hence there is no conflict between agents 1 and 2.

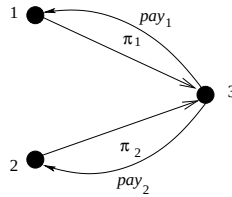


Fig. 4. Non-cooperative game, no conflict.

D. Non-cooperative, with conflict

This example is the classic and well-known Prisoner's Dilemma (fig. 5). Two agents promise to talk (or not talk) to their prison keeper (a third party). The prison keeper rewards both agents depending on the combined promises of both of the agents. The agents 1 and 2 are not in a position to cooperate (being held in different cells), but they affect one another's outcomes indirectly. This is a conflict because the actions of each player can have consequences for both of them, and it is non-cooperative because the two players cannot communicate.

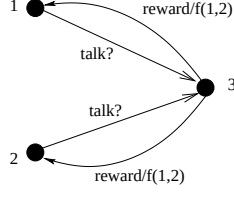


Fig. 5. Non-cooperative game, with conflict.

E. Constant-sum games

Let us now show that we can represent the special case of constant-sum (usually zero-sum) games, provided a certain level of cooperative infrastructure is in place. Such an infrastructure is often assumed in economic texts, but here we must be more careful. A constant-sum game is special because it implies an ability for one agent to control the value received by another agent. In a world of complete autonomy, that is forbidden, since each agent makes its own value judgements. Hence, to understand such a scenario, we must secure certain agreements between agents to agree on a currency standard. Constant sum games are not dependent on centralization, but we present two versions of such a game with and without centralization, both of which however show the fragility of the concept.

Theorem 1 (Constant-sum game): A promise interaction that forms a constant-sum game requires a finite number (at least N) of agents (N players and at least one adjudicator which may optionally be taken amongst the players), and all agents must agree on a standard currency.

Proof: Consider N promise interactions of the form:

$$\begin{array}{ccc}
 n_i & \xrightarrow{\pi_i} & A \\
 A & \xrightarrow{r_i/\chi(\pi_1, \pi_2, \dots, \pi_N)} & n_i.
 \end{array} \tag{4}$$

Each agent n_i promises an adjudicator A some service or behaviour π_i , and the adjudicator promises, in return, a piece of a common reward r_i , subject to a constraint $\chi(\pi_1, \pi_2, \dots, \pi_N)$. We would like this constraint to be that the sum of the values perceived by the agents is constant, i.e. we would like

$$\chi = \sum_{i=1}^N v_i(A \xrightarrow{r_i/\chi(\pi_1, \pi_2, \dots, \pi_N)} n_i) = \text{const}. \tag{5}$$

However, each agent's valuation function is its own private estimate and is unknown to A , so the best it can do is to arrange for

$$\chi = \sum_{i=1}^N v_A(A \xrightarrow{r_i/\chi(\pi_1, \pi_2, \dots, \pi_N)} n_i) = \text{const.} \quad (6)$$

Eqns. (5) and (6) can only agree if each agent n_i makes an agreement to use the same valuation $v_i = v_A, \forall i$, implying that the agents have a common currency. In a centralized framework, this implies an agreement of the form

$$\begin{array}{ccc} A & \xrightarrow{v_A} & n_i \\ n_i & \xrightarrow{U(v_A)} & A \end{array} \quad (7)$$

between each node and the adjudicator. Hence the payoff matrix for the game is:

$$\begin{aligned} & U_i \left(v_i(n_i \xrightarrow{\pi_i} A, A \xrightarrow{r_i/\chi} n_i) \right) \\ &= U_i \left(v_A(n_i \xrightarrow{\pi_i} A, A \xrightarrow{r_i/\chi} n_i) \right) \\ &= U_A \left(v_A(n_i \xrightarrow{\pi_i} A, A \xrightarrow{r_i/\chi} n_i) \right). \end{aligned} \quad (8)$$

The adjudicator node has the role of book-keeper. There is no impediment to identifying A as any *one* of the n_i agents, the conditions are binding and sufficient regardless of which agent plays the role of A . Similarly, agents with the role of A can be repeated any number of times. ■

The issue of who elects the adjudicator of the game (game-master) is still open, and we shall not discuss that further here.

In the proof above, we made use of a centralized construction. This is not strictly necessary for the currency. It is sufficient that every agent or player participating in the game should share a common knowledge of the valuation standard for currency and obeys a constant sum rule. The knowledge of the currency standard can spread epidemically through a promise graph to achieve the same result, provided the agents all abide by the same rules. This requires a special source node for the standard, which can be identified as A , so there is still a adjudicator node. One can argue about whether a cooperative group agreement is needed to guarantee such behaviour, i.e.

$$\{n_i\} \xrightarrow{C(v_A)} \{n_i\}. \quad (9)$$

this seems to obviate the need for a source node like A , but this is illusory, since this kind of fixed-point condition needs to be seeded with a boundary condition to make it converge. The nodes must however subordinate themselves to a set of rules that preserves:

$$\sum_{\pi_i, i, j} v_i(n_i \xrightarrow{\pi_i} n_j) = \text{const.} \quad (10)$$

This cannot be done without relinquishing part of their autonomy. Turning the argument around, it is likely that constant sum games do not represent a realistic interaction of autonomous agents.

F. Multiple strategies and mixtures

We have tacitly identified the promise of actions with strategies in the interpretation of a game in normal form. A promise becomes an alternative in the expectation of rational play. However, there are subtleties about actions and promises to be addressed. What if, for instance, two promises are mutually exclusive? What happens when we form mixed strategies? There are two ways we can envisage mapping games onto collections of promises:

- Strategies are binary: either we keep a promise or do not keep a promise.
- Strategies are multiple: distinct alternative promises for future action, some of which might be mutually exclusive.

Consider the game matrix for a two-agent game below, with payoffs..

(1,2)	π_1	π_2	π_3
π_4	(10,5)	(4,6)	(3,2)
π_5	(0,5)	(7,2)	(5,4)
π_6	(10,5)	(6,6)	(8,3)

The normal solution concept for such a game is to search for an equilibrium configuration of strategies for the players. Generally, such equilibria required mixed strategies to balance their books. However, this leads to a difficulty in interpretation for promises. Some promises might be mutually exclusive, some promises might break other promises, etc. Thus we cannot simply search for a numerical equilibrium for this matrix without applying the logic of promises to the problem also. The formal ambiguity can be dealt with by making promises conditional on certain prerequisites. However, there will be many games represented in the literature that assume information channels that are unavailable to autonomous agents. Readers should therefore take

care to remember the implications of complete autonomy. In particular, two agents can only be assumed to share common knowledge about π if they agree[2]:

$$\begin{array}{ccc} n_1 & \xrightarrow{\pi} & n_2 \\ n_2 & \xrightarrow{U(\pi)} & n_1. \end{array} \quad (11)$$

The issue of mixed strategies is something that must be addressed in both of the cases above. In a zeroth-level approximation to promise theory, one generally makes a default assumption that promises are always kept. This allows one to discuss the logic of promise graphs. However, as soon as one considers the matter of observation and repeated trials, and takes seriously the issue of why agents would keep promises, this default assumption becomes too simplistic to model real situations. We discuss some of these matters in the next section. For the remainder of the paper, we restrict our discussions to two-person games (or games that are two person factorizable).

VII. REPEATED BARGAINING: CONDITIONAL PROMISES

A dilemma that is faced by agents is whether to favour a long term benefit with a certain risk, or short-term guaranteed rewards by choosing a policy towards neighbours. If an agent only interacts once in its lifetime with another, it has nothing to lose by betraying the trust of the other. However, if there is the possibility of a future reprisal against it, the reckoning is a different one[13].

Agents have the possibility of *trading* promises of different types using a valuation based on an agreed measure of common currency, as pointed out in ref. [2]. This allows them to secure long term relationships with one another in a way that secures stability of interaction. Selfish agents will only predictably continue to interact if they receive some net payoff from the interaction.

However, in a real world pervasive computing scenario, one must be careful in attributing the failure to comply with promises to non-cooperative behaviour, as one might in economics. There are several reasons why an agent might fail to live up to its commitments in a cooperative relationship:

- Environmental uncertainties beyond the control of the agent intervene (noise).
- Unwillingness to cooperate emerges (change of policy).

- The failure of a dependency agreement, which invalidates a conditional promise (a third party spoils it).

This is not an extensive list, but each case is significant and important to the fragility of developing a regional policy in the ad hoc assembly of individuals.

So, given short term certainty versus long term risk, how might agents choose to behave? Suppose they surmise an incentive to cooperate:

- 1) Under what conditions would cooperation emerge and be stable in a group of selfish-minded individuals?
- 2) What kind of *behaviour* is it most beneficial for each individual to choose (strategy)? i.e. How does one harmonize selfish interest with common goals?
- 3) What might be done to promote the emergence of cooperation in a group by altering their policy for interaction?

VIII. THE GAME THEORETICAL VIEW OF COOPERATION

Cooperation between autonomous agents is voluntary and game theory has a number of attractive qualities for describing this situation. The preferred solution method of such game theoretical problems is the concept of Nash equilibrium[12]. An equilibrium is the end result of a ‘tug of war’ between players. Game theory therefore embodies the notion of *stability* in the result of the contest.

This is an appropriate model for management. The restriction to two players simply means that, in a graph of interactions between individual agents, each negotiation is made between individual pairs, without reference to third parties. This is also what is appropriate for an autonomous environment. We recall some basic notions from game theory.

Definition 5 (Two person iterated game): A game consist of two *players* or agents, which interact and make choices. For each choice i , by player a , and counter choice j by the other player, they each receive a payoff, which is determined by the *payoff matrix* Π_a with components $[\Pi_a]_{ij}$ (see fig. 6). The game can consist of one or several *rounds*. A game of several rounds is called an *iterated* game.

	C ₁	D ₁
C ₂	(R,R)	(S,T)
D ₂	(T,S)	(P,P)

Fig. 6. The payoff matrix $\Pi_{(1,2)}$ representing rewards for a single move in a dilemma game. The components $[\Pi]_{ij}$ represent the four quadrants for $i, j = 1, 2$

The benefits or losses each agent gets as a result of the interaction, are shown in the one-shot payoff matrix. In other words, the matrix implicitly gives a numerical description of the possible risks of an agent, and also what the agent can get away with in a relationship to another agent.

This matrix form of the game represents what happens during *one* encounter between two agents. If they should meet again, and the previous encounter might have influence on how the agents act next time they meet, this is modelled by several *iterations* of the game. Each iteration hence represents one interaction between two agents, and this interaction could be defined in several ways.

Definition 6 (Move and strategy): The choice of action $\vec{M}_{a,i}^T \in \{(1, 0), (0, 1)\} = \{\vec{C}, \vec{D}\}$ made by agent a in round i of a game is called a *move*. These concern keeping or breaking a given promise to the other player. For historical reasons, this is referred to as ‘cooperation’ (C) and ‘defection’ (D). A complete sequence of choices $\{\vec{M}_{a,i} | \forall i\}$ for agent a takes over the entire game, constitutes the *strategy* of that player.

The vector notation for moves is made with regard to fig. 6 in order to interface with the definition of the payoff matrix. Moves are in fact micro-strategies in the interpretation of the dilemma game as a normal-form, strategic game with constant payoffs. To avoid confusing micro- and macro-strategies, or extensive and normal form games, we use these terms: move and strategy as defined.

To account for the effects of a receding history, the total payoff of a player in an iterated game is generally a discounted sum of the payoffs in the individual rounds[14], [15]. Let $M_{a,r}$

be the move made by agent a in the r th round of the game, and let Π_a be the payoff matrix in this round for agent (player) a , given a counter-player $\bar{a}$, then over m moves,

$$\Pi_{\text{total},a} = \sum_{r=1}^m \delta^m (\vec{M}_{a,r}^T \Pi_a \vec{M}_{\bar{a},r}). \quad (12)$$

The factor $\delta \leq 1$ is called the discount paramter. One normally imagines oneself fixed at the present and looking out into (i.e. predicting) the future, based on past experience, so that m recedes into the future. What happens in the distant future (or distant past) is less important than what is immediately upon us, so its importance to the payoff is reduced in weight, or *discounted*. This duality between past and future is the same as that in all probabilistic theories in which evidence from the past is used as a model for the future.

The dilemma game has proven to be a very powerful tool for describing cooperation. The goal of this method is typically to find stable *solutions* to the iterated game, by balancing short and long term gain against one another.

The iterated game is played for a given number of rounds and the most profitable player, either during one round, or in the long run (several iterations) wins. The actual winner of a game is not really of interest to us in policy based management. What is important is the equilibrium outcome, i.e. which strategies or choices of promises agents choose to keep. To be able to predict the outcome of such an iterated game, one needs:

- The strategies of the other players involved in the game
- The number of encounters/interactions with the other players
- The payoff/reward for the different combinations of encounters
- The start state.

For our scenario of autonomous agents, this translates into finding out what kind of behaviour is most beneficial for an agent in the long run, dependent on what kind of environment the agent acts in, how often each agent meets agents with certain behaviour and so on.

Which strategies that are successful, will certainly depend on *all* these parameters. To predict the outcome of a simple game model might be completely impossible when the space of possible values for these parameters grow beyond certain sizes, for instance when the number of agents grows, and behavioural patterns vary.

For this reason, the method of *simulations* is commonly used for these kind of tasks. This means that we do not calculate an *absolute* static solution to the game, but rather vary the

condition, and examines the output for the different scenarios. In this manner, we look for behavioural patterns that turn out beneficial for each agent independently of the environment.

IX. PROMISES: CONSTRUCTING POLICY

Consider a set of agents or n_k , where k runs over an index set I . We interpret a promise, made by an agent n_k , as a declaration of intent to behave in accordance with a policy P_k , which is private to that agent. In other words, a promise is a unit of policy based management. We would like to know: in what sense is a single promise a stable or reliable quantity? And furthermore, when would collections of promises made between interacting agents form stable and reliable structures? How shall we model this?

It is clearly plausible that there is a connection between the two person games of the theory of cooperation and the likelihood that promises are kept between individuals in a pervasive computing environment. We shall now show this.

A promise graph $\Gamma = \langle I, L \rangle$ is a set of nodes I and a set of labelled edges or links between them L . Each directed edge represents a promise from one agent to another. The set of links L falls into several non-overlapping types, representing different kinds of promise. Thus each promise begins with a label π_i , denoting its type, and with a specification of the promise constraint, which it promises to uphold.

Lemma 1 (Games are bilateral promises): Let a two person strategic game be represented by a pair of 2×2 matrices Π_a , and moves $\vec{M}_i$ where $n_i, i = 1, 2$ are the agents and $a, b = 1, 2$ label the choices to cooperate or defect in the payoff matrix, e.g. $[\Pi_i]_{ab}$. There is a mapping of each game between the players onto a pair of promises.

Proof: For the players in the game, introduce a promise of type π_a and a return promise of type π_b , which are traded in the manner of fig. 7a. Now let $\vec{T}(\cdot)$ be a ‘truth’ function that maps any promise into a one of the values $\{\vec{C}, \vec{D}\}$, indicating whether the promise is obeyed or not by the promising agent. The truth function may be associated with a move in a game, of a type that maps one to one onto the promise type, and hence may be written equivalently:

$$\vec{T}(n_1 \xrightarrow{\pi_i} A_2) \rightarrow \vec{M}_1 \quad (13)$$

$$\vec{T}(n_2 \xrightarrow{\pi_j} A_1) \rightarrow \vec{M}_2 \quad (14)$$

$$\left[v_a \left(n_1 \xrightarrow{\pi_i} A_2, n_2 \xrightarrow{\pi_j} A_1 \right) \right]_{ab} \rightarrow [U_i]_{ab} \quad (15)$$

where v_i is the matrix-valued valuation function for agent i , which interprets the value of the mutual relationship to agent i ; π_a represents the promise body, which is labelled by the types a, b . We must have either $(A_1, A_2) = (n_1, n_2)$ or $(A_1, A_2) = (n_3, n_3)$ to mediate the game. The proof is now trivial noting that the arrow is from promises *onto* games, by a policy determined function v_i . ■

Note that, although we can identify a game with a pair of promises, not all promises can be meaningfully viewed as games, as we see in section XII. Any reasonable function v_i can be used to defined the mapping from promise to game, since the function v_i is to provide a autonomous *valuation* of received promises by each agent n_i . The game can then be used as a prediction about the rational behaviour of the agents as they interact.

A consequence of this relationship between promises and games, for autonomous agents, is that promises are not a complete specification of policy. One must also specify a policy for stabilizing the cooperative relationship over time, i.e. a strategy for the entire iterated game. Thus, although promises only provide a snapshot in time, the development of the policy graph must account for the role of time. We shall not take this point any further in this paper.

Proposition 1 (Policy must include iteration strategy): A complete specification of policy for autonomous agents in iterative, relationships must include a game strategy over multiple moves, i.e. when to keep promises, given the circumstances of the agent. Since authoritative models are built from autonomous models with additional promises, this implies that even authority-based management includes an implicit game strategy.

We now wish to elaborate on this point.

X. (NON)-COOPERATIVE DILEMMA GAMES

A common approach for modelling and analysing cooperation is to use dilemma games from two-person game theory[14], [15]. Here two individuals, without strictly opposing interests, have the possibility of exploiting each other's willingness to cooperate for selfish gain. This is a scenario that has shown to fit several real-life cases[14], [15]. It is known as a cooperative dilemma.

In the dilemma game, each player can choose between two different moves, *cooperate* (keep promise) or *defect* (i.e. do not keep promise). The interpretation of the dilemma game symbols

is thus: T is the payoff for receiving services from another agent, while refusing to contribute; R is the reward two agents receive when both are cooperating; P is the inconvenience both agents experience when they both refuse to collaborate; and S is the inconvenience of servicing an agent who refuse to contribute anything back.

There are many possible outcomes, which depend on the relative sizes of these parameters. For the game model to qualify as a Prisoner's Dilemma, in the common usage of the term, certain requirements need to be fulfilled: (i) $T > R > P > S$, and (ii) $R > \frac{(T+S)}{2}$. These requirements ensure that conclusions associated with the game are true.

A player can win the maximum payoff by 'defecting' (failing to act according to its own promises) when the other player is 'cooperating' (acting according to its promises), but if both players are defecting, they both receive a lower score than if they both had cooperated. Constraint (ii) assures that mutual cooperation is more beneficial than alternating cooperation and defection between the two players.

What makes this game model interesting, is the controversy that even though the choice *defect* is *dominant*, both players will actually end up with a better result cooperating in the long term.

This model is completely analogous to a scenario of autonomous agents, looking to establish agreements (promises) between one another, in order to create a larger system voluntarily. There is a potential long-term reward associated with providing services for other peer agents, but avoiding a commitment could thus seem beneficial in the short run.

XI. ITERATION STRATEGIES AND TEMPORAL POLICY

Would a promise made in advance of an interval of time increase the likelihood of service cooperation between the hosts[16]? This would, again, surely depend on the penalty for non-compliance. Such considerations are important for modelling service level agreements etc. The suggestion is that actual bilateral agreements are required, with implementable penalties for stable relationships to form. This is a necessary, but not sufficient condition.

We model this situation with multi-move games in which one agent can follow the history of its interaction with another and *punish* previous defections from the other player by failing to behave cooperatively in return. This scenario implies that a strategy based on cooperation could be successful, given that the opponent's strategy rewards cooperation.

An example strategy, which has received much attention in the literature, is *tit for tat*. This has proven to be very successful in playing cooperative dilemmas with a good average performance. Tit for tat is a macro-strategy and it is defined as follows:

Definition 7 (Tit for tat): Let r label a sequence of rounds in a game, and let $\vec{M}_{1,r} \in \{\vec{C}, \vec{D}\}$ be the move taken by agent 1 in round r , etc. The tit for tat strategy is defined by:

$$\vec{M}_{1,1} = \vec{C} \quad (16)$$

$$\left. \begin{array}{l} \vec{M}_{1,r} = \vec{M}_{2,r-1} \\ \vec{M}_{2,r} = \vec{M}_{1,r-1} \end{array} \right\} r > 1. \quad (17)$$

Notice that the agents always start out *cooperating*, or obeying policy rules, and then do against the opponent what the opponent did on the *previous* move.

The success of this strategy, in dilemma games, is characterised by the fact that it gets high scores on average, even though it does not necessarily always win (i.e. it does not always maximize the utility of its interaction). This strategy has several beneficial qualities:

- Simplicity.
- Rewards cooperation, and hence performs well with its own 'twin'.
- Punishes defection.
- Forgiving, hence single defections from opponent players does not harm the game too much.

The disadvantage is that it does not exploit recovery from random noise well enough. It tends to end in fruitless reactive oscillations between agents, once they have started. A continuous generalization of the game has further instability problems[17].

What this means for a group of autonomous agents in a dilemma game, using tit-for-tat, is that they will perceive a high level of payoff compared to agents acting differently. This is a useful result, as behaving according to *tit for tat* means cooperating unless other agents refuse, which again mean that a large group of *tit for tat* agents again would imply a group where *all* agents cooperate all the time.

However, in terms of the stability of promised relationships, a tit-for-tat strategy is probably too susceptible to oscillation. Some thing is required to dampen out the oscillations that would ensue on random error. A small amount of noise to unleash oscillations throughout a peer network, in which the agents had dependent promises.

The real dilemma in these games is this: if an agent in a trading relationship does not retaliate to broken promises, it will allow itself to be exploited. However, if it does retaliate, it participates in the contributing to a policy instability in the network.

XII. GAMES AND THE ECONOMICS OF PROMISES

What is the complete relationship between games and promises? We need to establish how the strategies of individual agents in a network interact and affect one another. Some relationships are purely fortuitous, with no guarantees of receiving a payoff for their actions. Others interactions are games which are bound by a trade of mutually beneficial promises. Thus not all promises are games, but all dilemma or bargaining games involve promises. How does the probability of keeping a promise relate to the experience gained from iterative dilemma games, and the discount parameter?

Since the cooperative dilemma gives us a natural notion of stable equilibrium for autonomous agents, let us ask: in what sense is a promise a cooperative dilemma, and which agents fall outside this assurance? In [2] we illustrated three different groups of possible promise types. These were the Service promise types, the Collaboration promise types, and the Usage promise types.

Consider two nodes, node A and node B , which each represent an autonomous node. As described in [2], a promise $a \xrightarrow{\pi} b$ from a to b is represented in graph notation by a directed edge (arrow) from node a to node b , and the label on the edge defines the *content* of the promise.

Definition 8 (Policy dilemma): A bi-lateral promise between two nodes, in which cooperation is interpreted as *keeping* the promise, and act as promise, and defection is defined as not keeping the promise. In order to qualify as a bargaining game, both nodes must receive some utility from the bilateral arrangement.

Consider the images in fig. 7. Here we see a number of promise graphs, some of which are policy dilemmas and some of which are not. There are three basic categories of promise[1]:

- 1) In a service promise of the form $a \xrightarrow{s} b$, node b is the recipient of something of value (the promised service).
- 2) In a use-promise, $a \xrightarrow{U(s)} b$, node a promises node b that it will avail itself of a service s , which has been promised by some node. In this case, node b does not receive anything

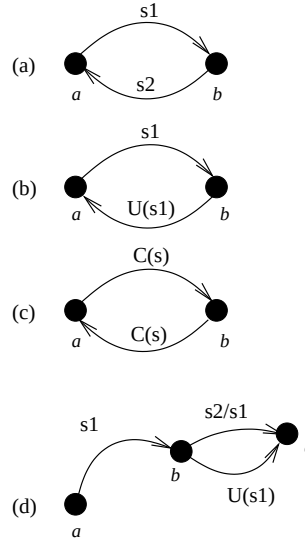


Fig. 7. Some 2 person games and their corresponding promises.

tangible, except an assurance that a will use the service it has been offered.

- 3) Similarly, in a cooperative promise $a \xrightarrow{C(s)} b$, node a promises node b that it will imitate b 's behaviour with respect to service s . Again, the only thing b receives is an assurance.

We must now ask: are all these dilemma games? Consider fig. 7.

In case (a) the two nodes exchange services or other similar promises of some value to one another. We may consider this interaction as representing a bargaining game between the players, if the promises are judged beneficial to the agents.

In case (b) however, the assurance of usage does not seem enough to bargain with. Indeed, node a says “if you do not give me s , I will not use it!”. This is hardly a bargaining position. One could similarly apply (b) to a cooperative promise: “If you do not give me s , I will not do as you do with regard to s ” (give myself s). Again, there are no grounds for bargaining.

In case (c), there is a game, since the two promises are on equal footing: “if you don't do as I do, I won't do as you do”. In both cases, the promises are of equal value and so they have a symmetry of mutual assurance, and we can define them to be a game. Indeed, case (c) can easily be shown to be a stable fixed point.

Case (d) contains no game either. Node c gives nothing in return to node b for its service s_2 ; node b offers nothing in return to node a for s_1 .

The services are promised without any trade taking place. In order to have a game, an agent must be able to exert an influence against its peers. It must have something to bargain with. There are two kinds of nodes that do not fit into this contract of stability.

Definition 9 (Opportunist): A network sink (a node with out-degree zero) is *opportunist*. It is vulnerable to loss of service, as it offers nothing in return. It is a non-profit node.

Definition 10 (Altruist): A network source (a node with in-degree zero) is an *altruist*. It is vulnerable and exploited, as it gives without taking anything in return. It is a ‘single point of failure’ for the dependent promises.

For example, we could repair the lack of economic incentive to cooperate in case (d) by modifying the graph to allow for some chain of payment (see fig. 8). This suggests a reason for introducing

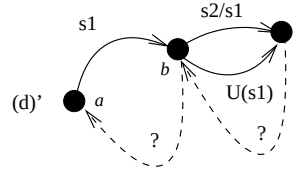


Fig. 8. An act of uncertain altruism can be turned into an incentivized bargaining game by assuring some payoff to the altruist.

a trading currency which is independent of that in ref. [2].

XIII. MINIMUM REQUIREMENTS

We can now propose some minimal requirements for sustainable promises, based in economic incentives.

Proposition 2 (Common currency): The need to bargain or trade with different currencies motivates an undirected graph of common currency, representing bilateral promises.

This graph is easily constructed from a knowledge of the promises alone. The common currency graph was exposed in ref, [2], from a different perspective. Having two independent motivations for such a graph is strongly suggestive of its importance, quite apart from the ability to model an underlying game. Indeed, the stability it engenders suggests a second principle for binding stability between the autonomous agents.

Proposition 3 (Principle of sufficient remuneration): Every agent should receive a ‘handshake’ promise in return for giving a promise. The handshake can be valued in any currency used by the agent.

This proposition can easily be proven if one posits equilibrium as a criterion for a well-formed policy. Given this principle, each interaction can be considered mutually beneficial to the agents concerned. If there are several promises between two nodes, then the combined value of the promises between any pair is the value that counts for their bargaining.

Example 3 (Binding agreements): The literature of games and economics often refers to ‘binding agreements’ in the sense of agreements that are enforced by some legal framework. In promise theory there is no such outside legal framework, unless one builds it from promises. Such a framework is very complex. A third party (government or law court) must promise to make laws known. All agents must promise to use the law. They must promise to make visible any contracts and practices they make to an arbitrator, who in turn must promise to inform law-enforcers of any breakage of laws. These in turn could enforce an isolation (incarceration) of an agent. To make an agreement binding by force, one must subordinate oneself to an outside force, or suffer large consequences. Such a scenario takes considerable infra-structure. This shows that commonly held assumptions can represent very complex scenarios (which is why criminals stand a chance of succeeding in breaking laws).

The idea of offering different interpretations of currency has been considered also by Piaget[19]. In his view, societies of agents bind into working constellations because they value one another by different criteria. Consider a transaction performed between two ‘agents’, represented by A and B , where A performs some action *on behalf* of B . Piaget assumes the following basic conditions, which differ from our starting point:

- A common scale of values for all agents.
- Conservation of values in time.

and remuneration values:

- r_A : *Renouncement value*. This value represents the *investment* made by agent A in order to perform an action on behalf of B .
- s_B : *Satisfaction value*. This value represents the satisfaction expressed by B as a result of receiving the action from A .

t_B : *Acknowledgement value*. This value represents the acknowledgement made by B to A as a result of receiving the action.

v_A : *Reward value*. This value represents the associated value of receiving the acknowledgement from B , as experienced by A .

Piaget distinguishes between so-called *real* and *virtual* values, where r and s are the real values, and t and v the virtual values in the transaction. The virtual values turn into real values when they are 'cashed in'.

XIV. STRUCTURAL STABILITY OF THE COOPERATIVE NETWORKS TO NOISE

A central issue is whether the cooperative behaviour is *stable* over some time interval. It is not reasonable to expect *continuous* stability during a larger time interval due to the interference of noise from random faults and environmental influences, but what about on average?

Example 4 (Compliance ranking): Consider a graph of n nodes, each making promises to a subset of the other nodes. Each promise is represented by an arrow between the node making the promise, and the node which is the receiver of the promise. In the result graph, the number of directed edges indicates how many *relationships* are established in the group, which also says something about how *clustered* the group is. Using graph theoretical terminology, we can compute the *connectivity* and the *clustering coefficient* of the group[20], [21], [22], [23].

Once the initial promises have been established, the interesting aspect is whether they will be kept. In a dilemma scenario, this translates into whether each node will cooperate or not over r iterations. After each iteration, each node's actions are evaluated, and it is determined whether the nodes cooperated or defected. If a defection has taken place, this is *recorded* in the nodes they defected against, and this can be used for the specific node's next choice of action.

If we assume all nodes are playing *tit for tat*, and the environment is without noise, it is possible to determine, without uncertainty, what choices each node will make in the next iteration. This means we can determine which nodes will cooperate, or how many will cooperate. However, this requires monitoring all separate occurrences, or all iterations, which is very resource-consuming, and not very effective for our purposes. So, what we choose to do, is to run all the r iterations, record all incidents, and then compute the *average probability* that each node will cooperate over the given time interval. The longer the time interval, the more useful is this number to predict

the cooperation rate for the group over time. This average or probability is simply the number of times a node has kept a promise, divided by the total number of iterations. This means that we assume actions being taken on all *iterations*.

The probability of whether each promise will be kept, is then a number in the interval $[0, 1]$, which in graph theoretical terminology is a *weight* on the corresponding promise edge. Having established this method of analysis, what is left is to define what we could achieve from the simulations. Our motivation is to find out under what conditions the probability of cooperation is as close to 1 as possible.

For two agents, we can easily predict the stable configurations under tit-for-tat by computing the spanning eigenvectors of the common currency promise graph. To do this, we write the strategy as an iterative matrix operator whose fixed points will be the stable solutions of the policy:

$$\lambda \begin{pmatrix} M_1 \\ M_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} M_1 \\ M_2 \end{pmatrix} \quad (18)$$

which has stable eigenvectors

$$\begin{pmatrix} M_1 \\ M_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix} \quad (19)$$

This shows that the two strategy modes that can persist in the graph are that both agents agree to agree (either keep their promises or not) for all time, or they enter into an eternal battle of opposites (cooperation and defection) for all time. In the absence of an initial condition (as in def. 7) there is no way to pick out which of the former is the case, but it seems reasonable to assume that promises will initially be kept if an agent bothers to make them. This method can be extended to larger graphs and we shall return to this matter in later work.

Random error in providing service, or signal noise, is inevitable in any real world computing scenario—a pervasive computing scenario, even more so, since systems are tightly interwoven with an unpredictable environment. What would be the effect of an agent suddenly being swamped by noise and being unable to meet its commitments? Would this unleash open warfare in the policy graph? Alternatively, what if a misconfigured or malicious agent were to start behaving in a non-cooperative manner, while engaged in promised relationships with other peers[24]?

Such a scenario can be modelled by the introduction of rogue agents in the graph, or using the eigenvector approach to find limiting configurations in which certain agents defect. The

interesting issue then is: will they be able to *exploit* other agents using other strategies like *tit for tat*, or will the cooperative group be able to ‘defend itself’? Contrarily, if an agent following the *tit for tat* rule enters an environment of agents following other retaliation policies, will the new agent be exploited, or is it possible for it to obtain the same or higher level of benefits in the new environment? Studies on these matters have been performed in the literature and one observes the following.

Definition 11 (Invasion and dominance): An new strategy is said to *invade* or *dominate* another strategy, if a *newcomer* gets a higher score with a *native* than a native gets with another native[15].

A native is an agent uses the same strategy as everyone else in an established group of autonomous agents. The concept of invasion is closely related to whether an environment can be stable independently of the new agents entering the group. Related to this, [15] defines the concept of *collective stability*.

Definition 12 (Collective stability): A strategy is said to be collectively stable if no other strategy can invade it.

It would be interesting to compare this to the eigenvector approach. A related concept related is *evolutionary stability*, due to Maynard-Smith[25]. A strategy is evolutionarily stable if a population of individuals using that strategy cannot be invaded by a ‘mutant’ adopting a different strategy.

In game theoretical terms, to evaluate the iterative stability of a strategy to the fullest, it is necessary to investigate the answers to the following questions: Can the strategy invade a population playing different strategies? Once established, is the strategy collectively stable? It is known that the discount parameter δ plays an important role in stability, often referred to as the ‘shadow of the future’. Our assertion is that this can be understood on the basis of general game theoretical experience[26], [20], [15], [18]. This knowledge motivates general principles for policy constraints, which can bind even lawless, autonomous agents into a social contract of cooperative behaviour. Such a contract will become increasingly harder to take for granted in pervasive environments.

XV. CONCLUSIONS

In this paper we have considered the relationship between promise graphs and the theory of voluntary ‘economic’ cooperation (see fig. 9). We have shown that, given the ability of agents to make their own valuations, there are promise configurations that form economic games. We propose that *sustainable configurations* are naturally predicted by the outcomes of these equivalent games. The body of work from cooperative game theory suggests that a complete view of a human-computer system policy, as an equilibrium of individual promises, requires two things: a common currency for motivating cooperation, and a stabilizing principle of remuneration (trading), that invokes bilateral promises to bind all nodes together. The fragility of the network is easily analyzed by methods of graph and reliability theory.

What does the game theoretical connection tell us about the probability that a promise graph will be a stable and lasting structure?

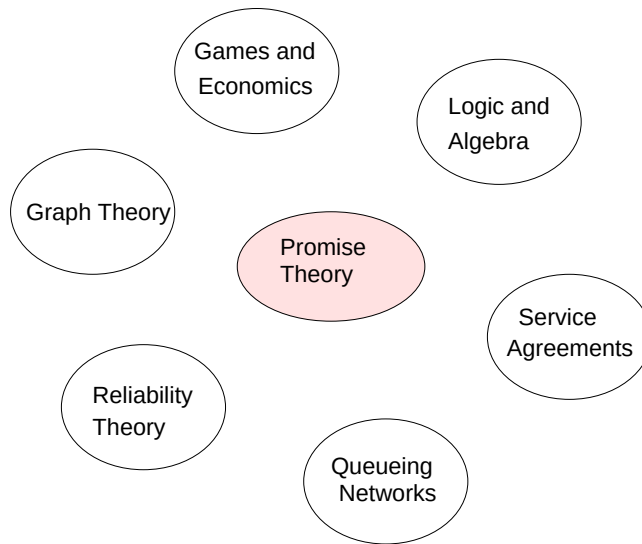


Fig. 9. An overview of how promise theory sits in the middle of a range of tools and brings them together in a simple way.

Naive cooperative game theory suggests policy strategies for maximizing the utility of an autonomous agent's policies.

- 1) *Initial cooperative behaviour*: When establishing a new promise or relationship, an agent should always act in a cooperative way, i.e. should keep the promise made to another agent.

- 2) *Punish bad behaviour*: If an agent fails to fulfil/keep a promise, this should be punished by the environment, so the agent lose some privileges.
- 3) *Reward good behaviour*: Agents that keep their promises should be rewarded.

This viewpoint is naive as we must also deal with defections that happen by accident: i.e. faults or ‘acts of God’. Immediate punishment is an efficient, but not necessarily desirable strategy, unless used as an incentive to motivate redundancy and fault tolerance of systems. Simulation leading to empirical conclusion is an avenue for future study.

Another vital feature of voluntary cooperation is the notion of recognition of the agents. The perceived effects of future reprisals must be sufficiently important to a node, in order to affect its behaviour. Thus promises must be *observable*.

According to the theory of cooperation, maintaining a selfish policy can lead to a stable environment of cooperative agents as long as certain requirements are satisfied: in particular, the future must be significantly important for each agent to be interested in keeping a promise. Thus long term relationships are conducive to stability. Ad hoc encounters are inherently less certain. A separate study of promise theory in ad hoc networks is the topic of a future paper.

Pervasive computing scenarios typically imagine an extremely dynamic environment, where the number of agents involved changes. Also, they are assumed to be very *heterogeneous*, which would imply a wide variety of different promises. Understanding the global impact of this world requires a common currency or economic model by which to convert and trade promises into another. The full picture of what happens in these scenarios is not trivial, but our framework suggests a way to make it possible from the common currency transformation. See also ref. [2].

In ref. [2], we alluded to a strategy of simplifying the discrete detailed view by replacing it with a common currency graph, in which each agents’s specific promises are collapsed into an overall view of its reliability. The numerical value translates into weights on the edges of the graph. There are clearly different ways of determining the edges on these graphs.

- *Estimated edge values*: One possibility would be to simply guess a probability that pairs of agents will keep promises of various types. This could be based on observation, unsupervised learning, or even simulation. However, this requires a knowledge of some kind of history of actual agents.
- *Negotiated edge values*: Another possibility, which captures the essence of a system with less initial input, is to consider a worst case scenario in the form of a bargaining game.

There are many threads to pick up on. We hope to report on these in future work.

Acknowledgement: M.B. is grateful to Professor Jan Bergstra for both valuable and promising discussions. S.F. would like to thank Kyrre Begnum for discussions. This work is supported by the EC IST-EMANICS Network of Excellence (#26854)

REFERENCES

- [1] Mark Burgess. An approach to understanding policy based on autonomy and voluntary cooperation. In *IFIP/IEEE 16th international workshop on distributed systems operations and management (DSOM)*, in LNCS 3775, pages 97–108, 2005.
- [2] M. Burgess and S. Fagernes. Pervasive computing management: A model of network policy with local autonomy. *IEEE Transactions on Software Engineering*, page (submitted).
- [3] M. Wooldridge. *An Introduction to MultiAgent Systems*. Wiley, Chichester, 2002.
- [4] J. Ferber. *Multi-agent Systems: Introduction to Distributed Artificial Intelligence*. Addison Wesley, 1999.
- [5] M. Burgess. A site configuration engine. *Computing systems (MIT Press: Cambridge MA)*, 8:309, 1995.
- [6] M. Burgess. Cfengine www site. <http://www.cfengine.org>, 1993.
- [7] S.A. Kripke. Semantical considerations in modal logic. *Acta Philosophica Fenica*, 16:83–94, 1963.
- [8] A. Lim, V. Srinivasan, and C-K Tham. A comparative study of cooperative algorithms for wireless ad hoc networks. In *Unpublished*, pages 1–8.
- [9] W.B. Norton. The art of peering: The peering playbook. Technical report, Equinix.com, 2001.
- [10] W.B. Norton. Internet service providers and peering. Technical report, Equinix.com, 2001.
- [11] E. Rasmusen. *Games and Information (Third edition)*. Blackwell publishing, Oxford, 2001.
- [12] R.B. Myerson. *Game theory: Analysis of Conflict*. (Harvard University Press, Cambridge, MA), 1991.
- [13] J.F. Nash. *Essays on Game Theory*. Edward Elgar, Cheltenham, 1996.
- [14] R. Axelrod. *The Complexity of Cooperation: Agent-based Models of Competition and Collaboration*. Princeton Studies in Complexity, Princeton, 1997.
- [15] R. Axelrod. *The Evolution of Co-operation*. Penguin Books, 1990 (1984).
- [16] J.D. Carrillo and M. Dewatripont. Promises, promises. Technical Report 172782000000000058, UCLA Department of Economics, Levines’s Bibliography.
- [17] K. Begnum, M. Burgess, T.M. Jonassen, and S. Fagernes. Summary of the stability of service level agreements. In *Proceedings of International Policy Workshop 2005*.
- [18] K. Begnum, M. Burgess, T.M. Jonassen, and S. Fagernes. On the stability of service level agreements. *IEEE eTransactions on Network and System Management*, submitted 2005.
- [19] J. Piaget. *Sociological Studies*. Routledge, London, 1995.
- [20] M. Burgess. *Analytical Network and System Administration — Managing Human-Computer Systems*. J. Wiley & Sons, Chichester, 2004.
- [21] V.K. Balakrishnan. *Graph Theory*. Schaum’s Outline Series (McGraw-Hill), New York, 1997.
- [22] C. Berge. *The Theory of Graphs*. (Dover, New York), 2001.
- [23] D.B. West. *Introduction to Graph Theory (2nd Edition)*. (Prentice Hall, Upper Saddle River), 2001.

- [24] X. Qie and S. Narain. Using service grammar to diagnose bgp configuration errors. *Proceedings of the Seventeenth Systems Administration Conference (LISA XVII) (USENIX Association: Berkeley, CA)*, page 243, 2003.
- [25] J. Maynard-Smith. *Evolution and the Theory of Games*. Cambridge University Press, Cambridge, 1981.
- [26] A. Rapoport. *Two-Person Game Theory*. Dover, New York, 1966.