Flexible Behaviour Regulation in Agent Based Systems

Michael Luck, Lina Barakat, Jeroen Keppens,
Samhar Mahmoud, Simon Miles, Nir Oren, Matthew Shaw, and Adel Taweel

Department of Computer Science, King's College London
London WC2R 2LS, United Kingdom
michael.luck@kcl.ac.uk

Abstract. Just as in human societies, for which we have developed *reasonably* effective systems to organise and manage interactions in such a way as to minimise the impact of erroneous or malicious behaviour, we also need to find ways to organise and manage computational entities in order to mitigate their potential deleterious effect on computational systems. In this paper, therefore, we analyse the role of trust, organisations and norms in a motivation-based view of agency that seeks to regulate behaviour, and illustrate some of these issues with aspects of several projects, including the CONTRACT project, concerned with electronic contract-based e-business systems.

1 Introduction

As information and communications technologies have progressed, there has been a change of focus from individual standalone computers to large-scale interconnected and open distributed systems. In fact, to a large extent, this move has already occurred, with such interconnectedness and openness becoming increasingly prevalent. While the benefits are myriad, for example in enabling dynamic service composition and virtual organisations, and supporting developments contributing to the realisation of ambient intelligence and Grid computing, there are also some important potential problems. In particular, little consideration has been given to problems analogous to those in human societies, where we need to consider the issues surrounding the use of regulations and their absence, of opportunistic and malicious behaviour.

Just as in human societies, for which we have developed *reasonably* effective systems to manage and organise interactions in such a way as to minimise the impact of erroneous or malicious behaviour, we also need to find ways to organise and manage computational entities in order to mitigate their potential deleterious effect on computational systems. While some work has been done on each of these concerns, their combination in large-scale open systems has not been addressed, yet this is a fundamental requirement if the visions of Grid computing and ambient intelligence, for example, are to be realised.

In this paper, therefore, we focus on the need for flexible behaviour regulation for open agent-based systems, which must be designed with a focus on techniques that anticipate and respond to the potential for erroneous or malicious behaviour. This requires an ability to reason about complex overall system operation, resulting from individual agent interactions, in support of mechanisms for control and management of systems

C. Guttmann, F. Dignum, and M. Georgeff (Eds.): CARE 2009/2010, LNAI 6066, pp. 99–113, 2011.

as a whole. We argue that this, in turn, demands an understanding of how motivations, organisations, norms and trust (among others) relate to each other, and how, in combination, they may give rise to effective and efficient systems.

The paper begins in Section 2 by reviewing a framework for characterising societies or systems along dimensions involved in flexible behaviour regulation.Then, in Sections 3, 4 and 5, it expands on the role of, and requirements for, organisations, norms and trust in such systems. Section 6 then introduces two distinct example applications, both driven by real-world needs, illustrating how the different aspects of the framework are relevant. Finally, the paper ends with broader conclusions.

2 Motivations, Organisations, Norms and Trust

As has been articulated elsewhere [16], much of computing, especially AI, can be conceptualised as taking place at the *knowledge level*, with computational activity being defined in terms of what to do, or goals. In this view, an agent's concern is taken to be the task of determining which actions to take to bring these goals about, yet the underlying motivation behind these goals, the *why* rather than the *how*, is not often considered. Despite being neglected, such motivation can have important and substantial influence over the way in which goals are achieved. As is generally accepted, *goals* specify what must be achieved but do not specify how; in this sense, they allow individual agents to decide for themselves how best to achieve them. Although this provides freedom in problem-solving, it provides little by way of direction or meta-level control that may be valuable in determining how best to achieve overarching aims. *Motivations* address this both by providing the *reasons* for the goal, and by offering *constraints* on how the goal might best be achieved when faced with alternative courses of action [15].

In this context, motivations are the starting point for considering flexible behaviour regulation, since they characterise the nature of the agents involved: at extreme points, whether they are malevolent or benign. Thus agents may be well integrated members of a system or society, cooperating with others when requested to do so, participating effectively in joint ventures, and contributing to the good of the whole. Alternatively, they may be malicious, seeking to take advantage of others, by requesting cooperation but not providing it, by taking the benefits provided by a society without contributing to its success, or they may simply be incompetent or unable to deliver.

Depending on the nature of such motivations, various mechanisms may be required to ensure effective system operation. For example, in cases where there is a prevalence of benign behaviour from individual agents, resulting from their motivations, there is less risk in interacting with agents because they will generally seek to cooperate without malicious intention. There is thus less risk in trusting others, since *defection* (which occurs when agents renege on their agreements with others) is unlikely; indeed, agents that are unwilling to trust others may miss opportunities for cooperation because of this. Moreover, in these situations, the use of excessive regulation through strict enforcement of system or societal rules (or *norms*) may hinder agent interactions to such a degree that cooperation is ineffective. Here, it is most appropriate to allow free reign to the full range of behaviours with little constraint, since malicious actions are unlikely, and the effort that might be expended on introducing stricter regulation may be wasted.

However, in cases where agents are less likely to be benign, some form of *behaviour regulation* is needed, either through constraints imposed by organisational structure and norms (limiting what is possible for agents to do) or through careful analysis of potential cooperation partners through an analysis of *trust and reputation*. (Here, trust is taken to be an individual's direct assessment of the reliability of another based on direct interactions with it, while reputation is an indirect assessment relying on the assessment of third parties, based on their interactions.) In the former case (when constraints are imposed by the organisational structure and norms), trust may be less important, since the system is heavily regulated through strict norms and enforcement. This is characteristic of the electronic institutions approach of several researchers (e.g., [6,8]) in which agents do not have the possibility of violating norms. However, despite this, if agents are less willing to trust others, then the possibility for taking advantage of opportunities in terms of cooperation may be ruled out. In the latter case, (when constraints are imposed by placing less trust in agents with poor reputation), we have a prevalence of agents with malicious motivations but their effectiveness is curtailed because of the care taken in determining cooperation partners. Here, if there is little organisational structure and lax enforcement of norms, there is a high likelihood that agents may defect, and since there is little protection from societal or system regulation, the role of trust is vital. Typically, agents should place very little trust in others in these situations.

In essence, this discussion characterises the axes of a three-dimensional space, described in [16], and illustrated in Figure 1, that identifies different *types* of systems or societies. More specifically, the x-axis represents motivations, with an increase in the value of x representing a prevalence of malicious motivations, indicating that agents are more likely to defect if they see more utility in alternative interactions. The y-axis represents organisational constraints, norms, and their enforcement, with an increase in value indicating the prevalence of stricter organisational structure, norms and enforcement. This can constrain the motivations of agents and prevent them from acting maliciously if they intend to do so. Finally, the z-axis represents trust, with an increase indicating an increase in the trust that agents place in others and, therefore, an increase in willingness to cooperate with others. In the figure, eight points in the space (at the vertices) are labelled by a circled letter, indicating different types of society or system.

As indicated above, societies A, B, E and F largely involve agents with benign motivations, so that the levels of organisational structure, norm enforcement and trust are less important, but societies C, D, G and H are more interesting. For example, C involves agents with malicious motivations, a lack of trust and lax organisational constraints, so that it is likely to be very inefficient. Society G uses strict organisational constraints to mitigate against these malicious motivations, society D uses a lack of trust to avoid potentially problematic interactions, and society H uses both. Each of these cases has been discussed in more detail above, but we have not yet provided a substantial consideration of the different axes of this space, which we do next.

3 Organisations

As computational systems increasingly comprise many tens or even hundreds of interconnected and interacting components, whether they be in a single physical location or

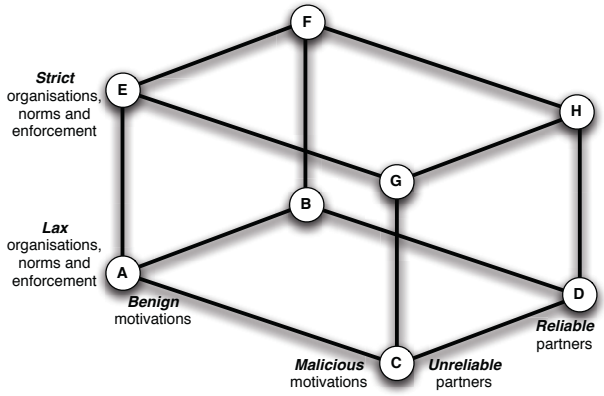

Fig. 1. The interplay between norms, motivations and trust (adapted from [16])

distributed across geographically diverse areas, they are increasingly difficult to manage in a manual fashion. One approach to overcoming these difficulties lies in the application of *organisational* models to structure the interactions between the components. Such organisations can be as simple as a hierarchy of authority or as complex as a society containing sub-society structures.

One of the most basic forms of organisational structure is a *hierarchy* [9], which naturally reflects structured relationships of authority or other subordinate relationships, for example. Hierarchies are very simple organisations, and are commonly applied in many fields because of their rigid structure and ease of understanding and use. In their most basic form, hierarchies are similar to trees, or directed graphs, in which there are nodes and edges connecting these nodes, but no cycles. Here, each node has an arbitrary number of child nodes connected to it via edges.

Although the obvious way to structure a hierarchy is by means of authority (with managers and employees or contractors in the business domain, and with commanders, sub-commanders and regular troops in the military), in fact this attribute could be something entirely different. In particular, it is this, whatever it may be, that determines the ordering of the hierarchy. For example, instead of authority, we might use time of execution as the ordering attribute in the case of a workflow hierarchy (e.g., [11]), which is a decomposition of a task into smaller sub-tasks, each node being a different sub-task. Such a hierarchy is needed because some of the sub-tasks may need to be executed before others can be started. Taking Figure 2 as an illustration of this type of hierarchy, if sub-task A must be executed before sub-task B, then A is the parent node of B. If sub-task C also cannot be executed until A is completed but it does not matter if B and C are executed sequentially or in parallel, then they may be siblings (that is to say that they are both children of the same parent). The direction of the edges shows the order in which the tasks must be completed, so in our particular application the root node must be executed first, followed by its children, and so on.

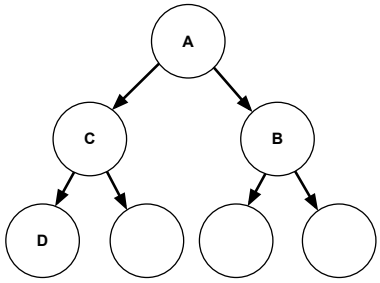

Fig. 2. A simple hierarchy

Similarly, as indicated above, we can have hierarchies that represent the authority of some entities over others, for example in the case of an employer's authority over an employee or a sergeant's authority over a private in the military. In this case, therefore, assuming the same graph from Figure 2, the direction of the edges reflects the order of importance or seniority in the hierarchy.

Importantly, such structures may provide constraints over interaction or communication. For example, one node in a hierarchy can only communicate with, and can only receive messages from, nodes to which it is connected. In Figure 2, node A can send and receive messages to and from node B, and *vice versa*. Conversely, A is not permitted to send and receive direct messages to node D, since they are not directly connected, and all communication must take place through the intermediary, node C.

Such structures can also aid control of a system. For example, in our hierarchy, the *apex* (root) node is the ultimate controller of the system, delegating tasks down the hierarchy. Nodes lower in the hierarchy send data produced lower down the hierarchy to inform subsequent decisions about the control of the system. In this way, the nodes in the system act as filters for communication: as data is passed up the hierarchy, each agent filters the data that is propagated upwards to ensure that no single node is overwhelmed with information. When decisions need to be made concerning information passed up the hierarchy, they are made at the lowest level possible, by the node that knows enough to be able to make the decision and has sufficient authority to execute it. In this way, hierarchies provide simple but effective structures with clear and simple protocols for communication.

Perhaps more interesting than simple static structures, however, are those that can change or adapt in response to prevailing circumstances. The goal of *self-organising* systems is to automatically form new organisation structures in situations when the existing structure is no longer optimal. However, systems of the complexity we are considering, comprising multiple interacting entities, can sometimes produce complex and unexpected global behaviour. This phenomenon is known as *emergence*, which refers to a novel *macro-level* situation that arises dynamically from the interactions between parts of the system at the *micro-level*. Here, the behaviours of the individual parts of the system can be fully specified or observed, but the interactions between these parts give rise to an unexpected and unpredictable macro-level behaviour.

4 Norms

More generally, the notion of a society can be taken to cover a group of individual agents, bound together in some fashion, through the adoption of an organisational structure, or rules to provide some structure arising from adherence to them. Such rules (also labelled conventions, social laws or *norms*) impose constraints on a population so that agents know both how they should act and what behaviour to expect from others. However, they are only valuable if used effectively; Fitoussi and Tennenholtz [7], for example, suggests that the balance between individual objectives and norms is critical, because norms must be sufficiently restrictive to have the desired effect for which they are applied, but must also be sufficiently flexible so that all objectives are equally feasible.

Along these lines, López y López and Luck [13] provide a more specific definition of norms within their normative framework. According to them, norms facilitate mechanisms to drive the behaviour of agents, especially in those cases where their behaviour affects other agents. Norms can be characterised by their prescriptiveness, sociality, and social pressure. In other words, a norm tells an agent how to behave (prescriptiveness) in situations where more than one agent is involved (sociality) and since it is expected that norms may often conflict with the personal interests of some agents, socially acceptable mechanisms to force agents to comply with norms are needed (social pressure).

4.1 Norm Enforcement

Since agents are autonomous, compliance with norms is not guaranteed, but violation, or non-compliance, of norms, can have negative effects on a society as a whole. As a result, there is a need to provide some means of encouraging compliance, which is typically achieved through the use of sanctions. In this view, the potential for sanctions to be imposed on a norm violator can be seen as norm enforcement, since agents must take into consideration the possibility of receiving some punishment if they violate norms. The issue of using sanctions to enforce norms has already been addressed, for example, by López y López et al. [14], who define the notion of *interlocking* norms. Here, two norms are interlocking if satisfying or violating one triggers the activation of the other. The first norm is the *primary* norm, and the second is the *secondary* norm. López y López et al. suggest applying the notion of interlocking norms to norm enforcement by specifying that reward or punishment norms are considered as secondary norms for the primary norms that the rewards or punishments are assigned to.

In this view, de Pinninck et al. [5] suggest the use of sanctions as a means of discouraging norm violation in Gnutella, a peer-to-peer file sharing application. Here, the system operates by different peers searching for files hosted by others and downloading them, and relies on the assumption that peers will both share and download. However, some may join the network and download files without contributing to the society by not sharing their own files. To prevent this problem of allowing peers to consume services without providing any of their own, de Pinninck et al. suggest adding a norm specifying that any agent that needs to download a file should share some files with others. If agents violate this norm, the violator is *ostracised* so that no other member of the society interacts with it and, as a result, it is denied access to the network's resources. In this

case, ostracising the violator is accomplished through the spread of negative reputation across the network.

4.2 Norm Emergence

From the description of de Pinninck's model, it is not clear if the norm itself is explicitly represented or merely implicit in the behaviour of agents as a result of attempting to generate the desired results. Along similar lines, Axelrod [3] proposes a game and undertakes experiments to illuminate the process of norm establishment (and norm collapse), when a norm arises through the guided behaviour of a group of agents. In his *norms game* and *metanorms game*, punishments are applied to agents who do not comply with norms, thus reducing their utility. However, because these punishments alone turn out not to be adequate, *metanorms* are introduced as *secondary* norms that help to enforce compliance of *primary* norms by applying punishments to agents that do not punish norm violators. Having integrated metanorms in his experiments, the results show that all runs end with a population that always complies with norms, so that norms always emerge when metanorms are used.

4.3 Norm Processing

In addition to these aspects, there are several other processes involved in the use of norms in open systems. Here, for completeness, we briefly outline the most important, including norm recognition, norm adoption and decision making. Norm *recognition* refers to the process by which an agent identifies a norm, or whether what might be taken to be a norm is, in fact, some other form of social input. For example, Andrighetto et al. [1] claim that the *norm recogniser* plays a very important role within the agent reasoning cycle in their EMIL-A architecture. When an agent recognises a new norm, it accepts it if it believes the norm concerns, or is directed towards, its behaviour. This is known as norm *adoption*. Conte et al. [4] state that an agent adopts a norm only if it believes that this norm helps, either in a direct or indirect way, to achieve one of the agent's goals. Based on this, the agent forms a normative goal that results from its decision to adopt the norm, but it does not make the decision to comply with this norm. Decision making is a critical phase of normative reasoning, as an agent decides within this phase if it is going to comply with a norm. Whatever the decision, it might have a major impact on behaviour. If the agent complies with the norm, some of its goals might conflict with the norm and, as a result, the agent will not be able to achieve any of these conflicting goals. On the other hand, if the agent violates the norm, then some punishments may be applied to the agent, which in turn can affect the achievement of some of its goals.

5 Trust

The use of norms in open computational systems can be extremely valuable, but without appropriate mechanisms to encourage compliance with them, they can become useless. This raises two important issues. The first, which is concerned with the nature of regulation in a society — how agents take decisions about norm compliance, and efforts

to encourage compliance, through enforcement and the severity of sanctions — is one we have largely outlined above. The second, which relies on the general levels of compliance in a society — in determining to what extent agents should be concerned that others may cooperate with them or defect (either because agents are more willing to defect in pursuit of more utility somewhere else or because there is uncertainty about whether agents can achieve the task) — requires a consideration of the third dimension of behaviour regulation, trust and reputation.

In particular, open multi-agent systems consist of a large number of interacting autonomous agents; each may have its own goals and may act to maximise its own benefit. Thus, in such environments, there is a challenge for agents to choose the most reliable interaction partner among many possible available. To cope with this challenge, many trust and reputation models have been introduced, enabling agents to calculate the trustworthiness of their potential partners, and then to choose the most trustworthy partner to interact with. By doing so, agents can maximise the chance that the interaction will achieve its potential benefits.

For example, TRAVOS [23] is a trust and reputation model for agent-based virtual organisations that computes the level of trust an agent (the truster) has in another agent (the trustee). More precisely, the trustworthiness of the trustee from the truster's perspective is the expected probability that the former will fulfil its obligations towards the latter in an interaction. The estimation of this expected probability is based on the outcomes of the previous direct interactions of the truster with the trustee, with each interaction being evaluated as either successful (the trustee fulfilled the agreed obligations) or unsuccessful (the trustee did not do so). Each calculated trust value is associated with a confidence level, which increases with the increased number of the observed outcomes.

When the confidence in the calculated trust value is below a pre-specified threshold, which means that there is a lack of personal experience between the evaluating agent and the agent being evaluated, the former depends on the latter's reputation to evaluate its trustworthiness. This reputation is obtained by combining the experiences (provided as outcome observations) the other agents had with the trustee and, as a result, the evaluating agent will have a larger number of observations with which to assess the trustee. Moreover, to cope with the problem of inaccurate third-party experiences (provided, for example, by misleading reputation sources), the truster estimates the *reliability* of each reputation source depending on the accuracy of its past opinions, and then uses this reliability value to decrease the role of unreliable opinions in the calculation of the trustee's reputation.

6 Case Studies

Given the above view of how we can understand the need for behaviour regulation in open agent systems, and the analysis of the relevant characteristics or dimensions that make up the space of such systems, we can move to consider how they are relevant in real cases. In this section, therefore, we introduce two case studies, one addressing a fully developed system, and the other addressing an application that would benefit from an analysis in terms of the concepts described in this paper. The first is concerned with

the need to provide contract-enabled e-business systems, to provide guarantees over service delivery between partners, based on several different motivating use cases [10], and illustrating the value of flexible behaviour regulation in normative settings. The second is focussed on a healthcare application for linking primary care providers to clinical researchers, showing the need for trust mechanisms and robust organisations in this domain.

6.1 Electronic Contracting

The CONTRACT project, created to explore multiple aspects of contract-based systems, developed a comprehensive framework for the creation, management [19] and monitoring [18] of electronic contracts. Here, electronic contracts are viewed as comprising sets of *norms*, i.e. statements of expected behaviour, usually expressed in terms of deontic concepts such as obligations, permissions and prohibitions. Development of the framework was informed by a series of case studies concerned with insurance settlement, aerospace aftermarkets [17] and certification testing, described by Jakob et al. [10].

The CONTRACT framework is a conceptual model for specifying applications using electronic contracting. The general architecture that was built using this framework provides an instantiation of relevant aspects of contract administration, through service-oriented middleware and multi-agent design patterns. Figure 3 illustrates the overall structure of the framework and architecture, which can be seen as a set of models and specifications, and comprising a methodology for adapting application designs to utilise electronic contracts. The primary component is the framework, at the top, which is the conceptual structure used to *describe* a contract-based system, including the contracts themselves and the agents to which they apply. Each level in the figure provides support for the components below it. Arrows indicate where one model influences or provides input to another.

Given a particular application, its framework specification enables other important information to be derived. For example, off-line verification mechanisms can check whether the contracts to be established satisfy particular properties, such as whether they can be achieved given the possible reachable world states. In turn, those states that are *critical* to observe during execution to ensure appropriate behaviour can be determined. In this context, critical states indicate whether, for example, an obligation is fulfilled or fulfillable, (e.g., achieved, failed, in danger of not being fulfilled, etc.). A state-based description, along with the deontic and epistemic implications of the specified contracts, can then be used to verify a system either off-line, using a model-checking approach [12], or with run-time monitoring [18].

Using the CONTRACT framework as a starting point, it is then possible to determine suitable processes for administration of the electronic contracts, including establishment, updating, termination, renewal, and so on. Such processes may include observation of the system, so that contractual obligations can be enforced or otherwise effectively managed, and these processes depend on the critical states identified above. Once suitable application processes are identified, we can also specify the roles that agents play within them, the components that should be part of agents to allow them to manage their contracts, and the contract documents themselves. An XML-based

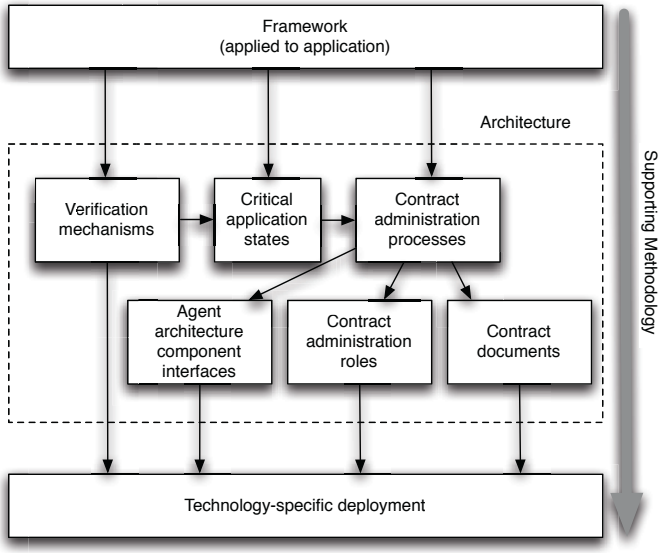

Fig. 3. The CONTRACT framework

language was developed to provide a concrete realisation of contract specifications using the framework, allowing contracts to be communicated and negotiated over [20].

All this provides the basic infrastructure required for establishing flexible behaviour regulation among agents or service providers in open systems, through a normative framework for agreeing contracts that impose structural relationships on the agents in the system, and through mechanisms for providing guarantees over their behaviour. In this context, their is an implicit suggestion that agents may not be benign, which is why contractual agreements are required.

Contracts. More specifically, the agreements between agents are formally described in electronic contracts, which document obligations, permissions and prohibitions (collectively clauses) on agents. Agents bound by contract clauses are said to be contract parties, and a contract specifies contract roles, which are fulfilled by contract parties, so that clauses apply to specific contract roles. Importantly, each of these clauses represents a norm, which can be interpreted here as socially derived prescriptions specifying that some set of agents (the norms targets) may, or must, or may not, perform some action, or see that some state of affairs occurs. As discussed earlier, norms can be understood as regulating the behaviour of agents: this is their role when encoded in contracts.

As indicated earlier, norms are social constructs, and it is meaningless to consider norms independently of their social aspect. This is because norms are imposed on the target by some other entity (the imposer) which must be granted, via the society, some power to impose the norm. Without this power, the norms target is free to ignore the norms prescriptions. With the presence of this power, a penalty may be imposed on an agent violating a norm. These penalties take the form of additional norms, giving

certain agents within a society permission to impose penalties (or obliging them to do so). This provides the means to establish the strict regulation that may be required without guarantees of benign behaviour. We assume that a contract is also made up of various descriptive elements, for example, stating which ontologies may be used to explain the terms found within it.

Since norms may have normative force only in certain situations, we associate norms with an *activation condition*. Norms are thus typically abstract, and are instantiated when the norms activation condition holds. Once a norm has been instantiated, it remains active, irrespective of its activation condition, until a specific expiration condition holds. When it occurs, the norm is assumed to no longer have normative force. Finally, independent of these two conditions is the norm's normative goal, which is used to identify when the norm is violated (in the case of an obligation), or what the agent is actually allowed to do (in the case of a permission). Obligations and permissions are the two norm types on which our framework focuses. Like others (e.g., [24]), we assume that additional norm types may be constructed from these basic types (e.g. a prohibition could be seen as an obligation with a negated normative goal). Norms may be activated, satisfied and discharged based on a number of factors including the status of other norms, the state of the environment (and the actions performed by other agents therein), and the status of contracts.

Contract Parties. Contracts in our system are agreed by agents, which are assumed to be autonomous, pro-active, flexible (decision-making) and social, and agents engage in contract-directed interactions to fulfil the clauses specified in a contract. Contract interactions require a minimum of two agents fulfilling the role of participants. Some applications may require contract-related processes to have certain properties, e.g. that violations are acted on, or that the integrity of the contract documents is maintained. These requirements lead to obligations on (and the creation and use of) administrative parties, and contracts may document their required behaviour. We thus have two distinct kinds of contract parties.

- *Business Contract Party Agents* for whom the contract is created: the obligations on the business contract parties are largely concerned with the business of the application.
- *Administrative Contract Party Agents* are required to ensure that the contract is accessible, retains integrity and legitimacy, is monitored and enforced, and other such administrative functions that ensure the contract has force. The obligations on these agents relate to their administrative roles.

Enforcement. Two particular administrative contract party roles are those of observer and manager [17]. The former detects whether the system enters a critical state (success, violation, in danger of violation) with regard to a particular clause. A manager reacts on the basis of observation, e.g. to inform a user of the problem, penalise a contract party in some way, and so on. There may be several observers and managers for an application, for example checking compliance on behalf of different users, and handling violations in different ways.

One of the important aspects of the CONTRACT architecture in ensuring and encouraging compliance with a contract is the process of *run-time monitoring* [18]. Here,

the key information available to third parties for this purpose consists of the messages exchanged between agents, and these messages are first gathered by *observer* agents. Monitors then receive the observations from observers that are explicitly entrusted by all contract parties to accurately report on the state of the world, and determine their status. The use of trusted observers ensures some degree of certainty that a norm is reported as violated if and only if it has in actuality been violated, and thus provides assurance that sanctions are only applied as and when appropriate. Once the status of a norm is ascertained through the monitoring process, the decision of what actions are to be taken is delegated to *manager* agents, which might apply sanctions for violations and rewards for fulfilment, as appropriate.

6.2 ePCRN

The ePCRN (electronic Primary Care Research Network)[1] is an infrastructure project that seeks to connect healthcare (primary care) with clinical research by facilitating the management of primary care clinical trials [2,22]. Clinical research is a vital resource for continually improving healthcare [21]. The issues involved relate to patient privacy and confidentiality, access to patient data across geographically distributed primary care clinics, heterogeneity of electronic health record (EHR) systems and interfaces, and so on, providing significant syntactic and semantic interoperability problems. The idea is that clinical researchers should be able to generate interoperable queries that can be submitted to all available clinics within the ePCRN network by adopting a Grid-based framework for distributed information access.

Clinical patient information is stored in EHRs in individual clinics or repositories (the record from several clinics combined into one repository), with other information potentially coming directly from hospital information systems (e.g., laboratory results, prescriptions, etc.). Each may have its own data format and interface. Researchers undertaking clinical trials need to find eligible patients by searching through this data, but preserving confidentiality and anonymity. As a result, they do not access the data directly at this point, and send requests to local clinics to invite patients to participate in studies, with only the local clinic being able to identify patients. Once patients are recruited, data for each participant is collected; if the EHR data is needed, it can be provided at this point, via secure or authorised access mechanisms. Finally, when the results of the trials are available, relevant data may be fed back into the patient record, but without repudiating the original unmodified record. A key challenge here is to enable access to patient data and maintain patient confidentiality; this requires a robust yet flexible confidentiality and security framework that enables healthcare providers to control their data sharing policy and benefit from clinical research output. Importantly, such a framework must both conform to organisational data and process flow and enforce system and local policy measures.

In the context of the three-dimensional space, it is clear that the organisational structure of, the relationships between, and the rules governing the researchers, the primary clinics, the practitioners and the patients, imposes significant constraints on what is possible. This is not simply a matter of efficiency and effectiveness, but is a requirement

[1] http://www.epcrn.bham.ac,uk

placed on the handling of data both legally and ethically. From a different perspective, the individuals involved must have strong trust in others to ensure that data is correct, that data will not be mis-used, and that patient records will not be modified inappropriately. At the same time, since all parties can be assumed to have benign motivations, and since the system is designed to impose the organisational constraints described, the levels of trust required are likely to be relatively high. Nevertheless, the relationship between the different axes described earlier in the paper should be clear.

7 Conclusions

In open agent systems comprising multiple autonomous agents, agent behaviour cannot be guaranteed, resulting in the possibility that overall system operation may be ineffective or inefficient. For this reason, there has been an increasing amount of research devoted to establishing appropriate mechanisms to encourage or enforce socially acceptable (and even valuable) behaviour on the part of individual agents. Drawing on analogous mechanisms in human societies, such mechanisms include those to establish, monitor and manage trust among agents that need to work together, based on experience of their previous interactions. However, in the absence of strong trust relationships to enable effective interaction in dynamic and open environments, some form of societal regulation must also be considered. In this respect, organisations and norms have been proposed as a means of mitigating the consequences of action when these trust relationships are absent.

In relation to these concerns, this paper has sought to elaborate on an earlier framework for characterising and contrasting different types of systems and societies as a means for understanding the requirements for behaviour regulation of the component agents. It considered systems with a prevalence of stricter organisations, norms and enforcement, those with agents inclined to behave maliciously, and those in which trust in agents is higher or lower. Given this analysis, the different dimensions of the framework were considered in more detail, and illustrated in relation to two distinct real-world case studies in which the issues raised are important factors in successful system design and development. The key message is that while the identified characteristics and techniques have been considered across a range of different research efforts, the relationship between them must also be analysed and understood in order to determine which techniques are appropriate in different circumstances.

References

1. Andrighetto, G., Campenni, M., Conte, R., Paolucci, M.: On the immergence of norms: a normative agent architecture. In: Proceedings of the AAAI Symposium on Social and Organizational Aspects of Intelligence (2007)
2. Arvanitis, T.N., Taweel, A., Zhao, L., Delaney, B.C., Peterson, K.A., Speedie, S.M., Sim, I., Weissman, J., Fontaine, P., Lange, C., Janowiec, M., Stone, J.: Supporting e-trials over distributed networks: A tool for capturing randomised control trials (RCT) eligibility criteria using the national cancer institutes (NCI) enterprise vocabulary services (EVS). Technology and Health Care 15(5), 298–299 (2007)

3. Axelrod, R.: An evolutionary approach to norms. The American Political Science Review 80(4), 1095–1111 (1986)
4. Conte, R., Castelfranchi, C., Dignum, F.P.M.: Autonomous norm acceptance. In: Papadimitriou, C., Singh, M.P., Müller, J.P. (eds.) ATAL 1998. LNCS (LNAI), vol. 1555, pp. 99–112. Springer, Heidelberg (1999)
5. Perreau de Pinninck, A., Sierra, C., Schorlemmer, W.M.: Distributed norm enforcement: Ostracism in open multi-agent systems. In: Casanovas, P., Sartor, G., Casellas, N., Rubino, R. (eds.) Computable Models of the Law. LNCS (LNAI), vol. 4884, pp. 275–290. Springer, Heidelberg (2008)
6. Esteva, M., Rodríguez-Aguilar, J.-A., Sierra, C., Garcia, P., Arcos, J.-L.: On the formal specification of electronic institutions. In: Sierra, C., Dignum, F.P.M. (eds.) AgentLink 2000. LNCS (LNAI), vol. 1991, pp. 126–147. Springer, Heidelberg (2001)
7. Fitoussi, D., Tennenholtz, M.: Choosing social laws for multi-agent systems: Minimality and simplicity. Artificial Intelligence 119(1-2), 61–101 (2000)
8. García-Camino, A., Noriega, P., Rodríguez-Aguilar, J.A.: Implementing norms in electronic institutions. In: Thompson, S., Pechoucek, M., Steiner, D. (eds.) Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 667–673. ACM Press, New York (2005)
9. Horling, B., Lesser, V.: A survey of multi-agent organizational paradigms. The Knowledge Engineering Review 19(4), 281–316 (2005)
10. Jakob, M., Pechoucek, M., Chábera, J., Miles, S., Luck, M., Oren, N., Kollingbaum, M., Holt, C., Vazquez-Salceda, J., Storms, P., Dehn, M.: Case studies for contract-based systems. In: Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems, pp. 55–62 (2008)
11. Kota, R., Gibbins, N., Jennings, N.: Self-organising agent organisations. In: Proceedings of the Eighth International Conference on Autonomous Agents and Multiagent Systems, pp. 797–804 (2009)
12. Lomuscio, A., Qu, H., Solanki, M.: Towards verifying contract regulated service composition. In: Proceedings of the 8th International Conference on Web Services (ICWS 2008), Beijing, China, pp. 255–261 (2008)
13. López y López, F., Luck, M.: Modelling norms for autonomous agents. In: Chávez, E., Favela, J., Mejía, M., Oliart, A. (eds.) Proceedings of The Fourth Mexican Conference on Computer Science, pp. 238–245. IEEE Computer Society, Los Alamitos (2003)
14. López y López, F., Luck, M., d'Inverno, M.: A normative framework for agent-based systems. Computational and Mathematical Organization Theory 12(2-3), 227–250 (2006)
15. Luck, M., d'Inverno, M.: Motivated behaviour for goal adoption. In: Zhang, C., Lukose, D. (eds.) DAI 1998. LNCS (LNAI), vol. 1544, pp. 58–73. Springer, Heidelberg (1998)
16. Luck, M., Munroe, S., Lopez y Lopez, F., Ashri, R.: Trust and norms for interaction. In: Proceedings of the IEEE International Conference on Systems, Man & Cybernetics, pp. 1944–1949. IEEE, Los Alamitos (2004)
17. Meneguzzi, F.R., Miles, S., Luck, M., Holt, C., Smith, M., Oren, N., Faci, N., Kollingbaum, M., Modgil, S.: Electronic contracting in aircraft aftercare: A case study. In: Berger, M., Burg, B., Nishiyama, S. (eds.) Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multiagent Systems, Industry and Applications Track, pp. 63–70 (2008)
18. Modgil, S., Faci, N., Meneguzzi, F., Oren, N., Miles, S., Luck, M.: A Framework for Monitoring Agent-Based Normative Systems. In: Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009), Budapest, Hungary, pp. 153–160. IFAAMAS (May 2009)

19. Oren, N., Panagiotidi, S., Vázquez-Salceda, J., Modgil, S., Luck, M., Miles, S.: Towards a formalisation of electronic contracting environments. In: Hübner, J.F., Matson, E., Boissier, O., Dignum, V. (eds.) COIN@AAMAS 2008. LNCS, vol. 5428, pp. 156–171. Springer, Heidelberg (2009)
20. Panagiotidi, S., Vazquez-Salceda, J., Alvarez-Napagao, S., Ortega-Martorell, S., Willmott, S., Confalonieri, R., Storms, P.: Intelligent contracting agents language. In: Behaviour Regulation in MAS, AISB 2008 Convention Communication, Interaction and Social Intelligence, pp. 49–55 (2008)
21. Sim, I., Olasov, B., Carini, S.: An ontology of randomized controlled trials for evidence-based practice: content specification and evaluation using the competency decomposition method. Journal of Biomedical Informatics 37(2), 108–119 (2004)
22. Speedie, S.M., Taweel, A., Sim, I., Arvanitis, T., Delaney, B., Peterson, K.: The primary care research object model (PCROM): A computable information model for practice-based primary care research. Journal of the American Medical Informatics Association 15(5), 661–670 (2008)
23. Teacy, W.T.L., Patel, J., Jennings, N.R., Luck, M.: Travos: Trust and reputation in the context of inaccurate information sources. Autonomous Agents and Multi-Agent Systems 12(2), 183–198 (2006)
24. von Wright, G.H.: Deontic logic. Mind 60, 1–15 (1951)