Norm-based agency for designing collaborative information systems

Kecheng Liu*, Lily Sun*, Alan Dix† & Mohan Narasipuram‡

*School of Computing, Staffordshire University, Stafford, ST18 0DG, UK, email: K.Liu@staffs.ac.uk, L.Sun@staffs.ac.uk, †Computing Department, Lancaster University, Lancaster, LA1 4YR, UK, email: alan@hcibook.com, and ‡Department of Information Systems, City University of Hong Kong, 83 Tat Chee Avenue, Hong Kong

Abstract. *Information systems are organizations in which signs are created, processed and consumed. In order to fulfil the organizational goals and objectives, its members must understand their responsibilities and authorities, and must act co-operatively. The key to this organization and co-ordination lies in norms, which define responsibilities and authorities for each human agent, and establish regularities of behaviour. In the context of co-operative work, where 'intelligent' software agents are involved, to understand the norms of behaviour of various human agents becomes critical. Software agents can perform some tasks autonomously on the user's behalf. Such delegation involves a set of complicated philosophical and legal issues. After discussion on delineation of various boundaries of responsibility and authorities, this paper addresses norms and normative behaviour of human agents within an organization. It discusses the taxonomies of norms and a method of norm specification, with examples. Finally it presents an approach of norm-based agency for designing collaborative information systems and a case study of an insurance claim for illustration.*

Keywords: Agency, collaborative information systems, normative behaviour, norms, semiotics, systems design

INTRODUCTION

Information systems are organizations where people, usually with the aid of technology, perform their duties and carry out business activities. These systems therefore are social systems. In business operations, signs are created, processed and consumed within organizations for social and economic purposes. In order to fulfil the organizational goals and objectives, 'actors' within an organization must act co-operatively, and the success of an organization depends on the work of all its members. In a business organization, rules exist and must be followed. Many rules may not be formally and explicitly defined, though they affect members of the organization implicitly; all these formal and informal rules can be called norms

(Wright, 1963). The co-ordination and co-operation are achieved by the norms that govern people's behaviour. Advanced information technology can support and facilitate distributed and dispersed work activities; such information technology includes computer-supported co-operative work. However, the design of an effective collaborative information system will require a sound understanding of the business organization, which further leads to a need of a co-design of business processes and computer-based systems. This position can be seen from many important literature and recent work in collaborative information systems (see, for example, Holt, 1985; Krogh, 1995; Schmidt & Simone, 1996; Teege, 2000). An effective approach to derive a conceptual design for collaborative information systems is norm-based theory (Stamper *et al*., 2000), which is elaborated in this paper.

In a large, distributed, network computer-based system, 'intelligent' software components are often used. Such software components are coined as 'intelligent agents' (Wooldridge & Jennings, 1995). This notion may have been potentially confusing with human agents in a social, business situation. Human agents in a business organization will act according to the social, organizational norms. They can be assigned duties and authorities, and can be held responsible for their actions. Many of these concepts have been adopted by the AI (artificial intelligence) research community. In their definitions of machine 'agency', characteristics such as autonomy, social ability and communication with other agents are thought to be fundamental. A software agent is supposed to be capable of expressing beliefs, desires and intentions. However, key questions have to be asked before the agency theory is applied to machines. Do we understand the patterns of human agents well enough? Are we able to capture and represent human beliefs, desires and intentions? How do we express human authorities and responsibilities?

This paper will address definition of human responsibilities and distribution of functions between human and machine agents. It discusses the types of norms and their functions in defining human responsibilities, followed by illustrations on how human functions can be delegated while authorities and responsibilities still remain with the human beings.

BACKGROUND

Mark Klein, Centre of Co-ordination Science at MIT, stresses the potential roles of intelligent software agents in collaborative information systems, and proposes that basically, 'agent = components PLUS co-ordination' (Klein, 1999). In a collaborative information system, people have business responsibilities and commitments, some of which can be delegated to software agents in many roles: proxies, performing actions and interacting with other agents on a user's behalf; mediators, passing on or routing messages; translators, modifying the form of messages; or co-ordinators, suggesting or determining allowable actions within the collaborative environment. From a semiotic standpoint we can see that most of these roles include aspects of sign processing or transmission. We therefore look to the software agents in a collaborative information system and examine how they help the human users in their work, while the emphasis of our work will not be placed on system component and architecture.

Agency in collaborative information systems

Collaborative information systems are composed of human users (human agents), the media through which they communicate and the objects on which they act (Dix *et al*., 1994). The objects are important both as the focus of work and as a means of communication through artefacts (Dix, 1994). However, for this paper we will concentrate on the agents and media. In addition, many collaborative information systems include software agents intended to improve the efficiency and quality of work. In fact, the distinction between agents in a software system and the media of communication is itself complex. Even in a physical medium such as the postal system, human agents (the postman, sorting office staff, etc.) act as mediators and routers of letters. The same may happen in a software system and, in addition, parts of the system labelled as agents may perform fairly trivial tasks.

Characteristics of machine agents

An *agent* is normally seen as a person who acts for another, especially one who looks after or represents the business affairs of a person or firm. In organizational semiotics, the term 'agent' denotes a person or a group of people in a social system who take responsibilities and perform actions (Stamper & Liu, 1998). Recently the concept of agency has been adopted in computing science, but used specifically to refer to a software component or an 'intelligent' machine as an 'agent'. The computing technology of an 'intelligent agent' has proved useful in many application areas, for example the Internet, communications, network management, electronic commerce and database management. It is also highly relevant for collaborative information systems because the number of beneficiaries is greater than in a single user situation, and hence the efficiency gain is much more significant.

Research workers tend to agree on the following characteristics necessary for a machine-based agent (Wooldridge & Jennings, 1995). *Autonomy*: an agent can operate without the direct intervention of humans or others, and has some kind of control over its actions and internal state. *Social ability*: one agent interacts with other agents (and possibly humans). *Reactivity*: it has the ability to perceive its environment (the physical world, a user, other agents, the Internet, etc.) and respond to changes in the environment. Finally, *Pro-activity*: an agent takes initiatives and behaves pro-actively in order to achieve predetermined goals.

Researchers in DAI (distributed artificial intelligence) claim that an intelligent software or machine agent has a way to exhibit an intentional stance, such as knowledge, belief, desire, intention, obligation, commitment, choice, and so on (Wooldridge & Jennings, 1995). It is believed that these complex notions can be captured and represented to the extent that automatic manipulation and reasoning can be performed on a rational basis. A great deal of effort has been put into representation of 'beliefs, desires and intentions' using modal logic, temporal logic and the like, e.g. Rao (1995). Success stories of machine agents can be seen in a number of applications areas. As reported by Smith & Mamdani (1996), they can be used to assist people with routine but important tasks, to undertake complex tasks rapidly, to act

as gophers on a human user's behalf, to explore large cognitive search spaces, to find elusive solutions and to deal with data tasks.

Warnings are given regarding issues such as responsibility, authority, liability and ethical questions (Eichmann, 1994; Nwana & Ndumu, 1996). These issues have not, in general, been put on the immediate research agenda; therefore, little attention has been paid to how to understand and handle the questions of obligation, commitment, responsibilities and authorities. It is argued later in this paper that these questions are important and have immediate relevance.

Norms and normative behaviour

Norms are developed through the practical experiences of people in a culture, and in turn have functions of directing, co-ordinating and controlling actions within the culture. A research group or a working team may have a subculture and therefore may have 'local' norms. The norms will provide guidance for members to determine whether certain patterns of behaviour are legal or acceptable within the given context. An individual member in the community, having learned the norms, will be able to use the knowledge to guide his or her actions, though he or she may decide to take either a norm-conforming or a norm-breaking action (Liu, 2000; Liu & Ong, 1999). When the norms of an organization are learned, it will be possible for one to expect and predict behaviour, and hence to collaborate with others in performing co-ordinated actions. Once the norms are understood, captured and represented in, for example, the form of deontic logic, this will serve as a basis for programming intelligent agents to perform many regular activities.

EMBEDDED NORMS IN ELECTRONIC MEDIA

Most collaborative information systems require some form of sign processing and transmission, often mediated by various levels of electronic media and computer agents. The communication over such media will be governed by norms of behaviour and communication, but with some intermediaries it becomes problematic where these norms are interpreted and understood. It is clear that a human translator working for a senior diplomat would not only use semantic knowledge as part of the language translation, but would also use knowledge of the cultural and diplomatic norms in choosing appropriate phrasing.

Mediation and agency

The most common form of electronic mediation is the telephone. Although there is extensive signal processing (probably digital) and routing of signals, the medium is semantically neutral. The affordances (Gaver, 1992) or constraints (Clark & Brennan, 1991) of the medium may modify human communication, but the norms are produced and interpreted by the human users alone.

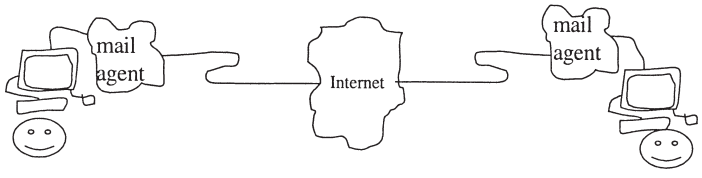

Figure 1. Communicating by email – agents and routers.

Email is slightly more complex. In the simplest case an email system consists of mail agents for each user and a transport medium (a computer network such as the Internet). The phrase 'mail agent' is used to refer to the actual program the user interacts with; that is, the user interface of the email system. These agents are important as they embody most of the 'intelligence' of an email system as a whole (see Figure 1).

Low-level email standards

Because email operates across platforms and between proprietary systems, there are extensive standards covering issues such as addressing, routing and message content. These standards can themselves be regarded as a form of low-level norm and have arisen by a variety of processes: international committees such as X400, proprietary standards such as MAPI and standards arising from agreed common usage such as the Internet standards. We shall look at the Internet standards in more detail to see where higher levels of norm are embodied. As we move up through the levels of the system more complex norms are embodied.

Consider an email message such as the one in Figure 2. At the lowest level, email is delivered from machine to machine using the simple mail transfer protocol (SMTP). The sending machine contacts the recipient or routing machine, tells it who the message is for (A.J.Dix@soc.staffs.ac.uk, counter-staff@bigbank.com, and D.Tracy@fbi.gov), and who it is from (K.Thomas@bigbank.com) and then sends the body of the message *including all the headers*. This level of protocol does not look inside the message at all. It is like addressing an envelope, putting a return address on the back, putting a letter inside, sticking it down and then posting it. The post office should not look inside the envelope, but simply deliver it. Note especially that the interpretation of headers is left entirely to the mail agents.

The headers themselves and the body are subject to standards governing various common headers with their meanings (From:, To:, Subject:, etc.) and the coding of different kinds of file format and media (MIME). The SMTP does not check that the recipients or sender agree with those in the body, or that the message is conformant with MIME or other standards. Because of this neutrality of the underlying medium, mail agents can use different encoding standards for email content and additional headers (normally prefixed with 'X-'). Of course, if email is sent between different kinds of email agents, parts of the message may not be properly interpreted, especially file attachments.


From: K.Thomas@bigbank.com
To: A.J.Dix@soc.staffs.ac.uk
Subject: ac. no. 765 183 0078
Cc: counter-staff@bigbank.com
Bcc: D.Tracy@fbi.gov

Dear Prof. Dix,
The state of your account is causing some concern. In particular, your recent withdrawal of £117,000 has considerably exceeded your overdraft limit of £500. I have therefore asked staff to refuse any further withdrawals and urge you to contact me immediately on your return to the country.

Yours,
Kate Thomas (Senior Account Manager)

Figure 2. Typical email message.

The message in Figure 2 has a 'To:' field, a 'Cc:' (copy to) field and a 'Bcc:' (blind copy) field. The email is delivered to the recipients in all these fields. As far as the email agent is concerned, when posting the message 'To:' and 'Cc:' are treated the same. However, it does recognize the different status of 'Bcc:' and before sending the email it removes the 'Bcc:' line from the header. That is, the email agent 'understands' one aspect of the norms governing email usage. Where then lies the difference between the 'To:' field and the 'Cc:' field? In fact, the difference lies in the way the recipients interpret them. If you are named in the 'To:' list, you will feel a greater need to respond than whether you are in the 'Cc:' list, when you may simply need to take note of the contents. The precise details will depend on the norms of the organizational culture of the sender and recipients (and may be misunderstood because of the neutrality of the medium). However, the important thing is that the interpretation of these norms is performed by the users themselves.

Rules and filters

The knowledge about 'Bcc:' is built into the email agent by the designer. Many email systems allow users to specify filtering, which may alert the user, file messages or even discard them according to simple rules such as:

if field 'Subject:' contains 'special offer'
then move to Trash

These rules embody individual norms of behaviour that are executed by the mail agent. However, such rules are limited by the restricted set of fields available. They allow simple decisions based on sender or recipient (mailing list or personal), but more complex decisions require text matching rules (as above) which are notoriously error prone. Also note that

participants in an email exchange may not even be aware that their colleague is using an automatic email filter.

Semi-structured message systems such as Group Lens (now known as Oval) offer a more explicit representation of the semantics of a message (Malone *et al*., 1987). These systems incorporate typed messages. Messages of a particular type will have different fields; for example, a seminar announcement may have a Speaker: and a Title: field. These fields allow more efficient personal filtering of messages based on message types and the fields they contain. However, this is based on the explicit co-operation of colleagues. Different types of message can easily be created and subclassed from existing types, but the efficiency of filtering depends critically on the common understanding and usage of types and fields. Thus, we have a system which can embody, and be tailored to, some of the group norms of particular communicative intra- and interorganizational subgroups.

Workflow

Some of the most explicit coding of norms within messaging systems is found in workflow systems (Kreifelts *et al*., 1991; Prinz & Kolvenbach, 1996). These embody predefined corporate norms of behaviour. Users are presented with documents/messages where they must fill in or approve specific parts; depending on their responses the document message will automatically be passed on to the next person who must see it, according to the predefined business process. For example, an employee may fill in an electronic expense form. On completion, the form is passed to the line manager for approval, then to the finance department who will assign a payment code and finally to payments who write the cheque. The popularity of business process re-engineering (BPR) has driven the widespread take up of such systems, often implemented using Lotus Notes or similar scriptable messaging systems.

An often quoted example of workflow systems is Coordinator (Winograd & Flores, 1986). This is a typed messaging system where the types, and rules relating to types, are based on speech-act theory (Searle, 1969). The workflow system is designed to support work process specified by norms. Generic patterns of 'speech acts' have been identified, most well known being the 'conversation for action' (CfA). According to Winograd and colleagues, these speech acts represent norms of conversation across societies. In Coordinator, the initiator of a conversation must identify what type of conversation is being performed and for each message the users must identify what illocutionary point (encoded in message type) it is intended to convey (a request, a refusal, etc.). The system allows messages that conform to the speech-act patterns. When exceptions to prescribed patterns of behaviour (i.e. breakdowns) occur, the system will rely on necessary human interventions.

Levels of embedded norms

Consider again the move from simple email through filtering and semi-structured messages, to workflow systems and Co-ordinator. First, this represents a gradual increase in the explicit

System	Embodiment of norms
simple email	naming
semi-structured messages	group norms
workflow	organisational norms
Coordinator	social and linguistic norms

Figure 3. Norms embodied in computer systems.

semantic content encoded within the messages. Second, this explicit semantic knowledge also represents a gradual shift in the communality of norms embodied within the mail agents (or messaging system) as illustrated in Figure 3.

In all these cases, the rules obeyed by the agents have been explicitly coded by someone. These rules embody different levels of norms, but can we say that the agents interpret or understand these norms, or are they merely obeying orders? Is it that the humans in the system impute understanding of norms to the computational agents? Of course, we may have human agents acting on our behalf (e.g. a stockbroker), whom we also expect to obey our orders. The main difference is that the computational agent obeys fixed deterministic rules, whereas the human agent is expected to act with discretion within parameters and must thus understand as well as be able to obey the norms. However, it is important to be able to specify these norms, otherwise how can we be sure that a human agent is acting within the appropriate boundaries of responsibilities and authorities?

NORMS AND RESPONSIBILITY

Still, difficult issues arise in twofold: how to decide the extent to which human responsibilities should be delegated to a software agent, and what are the roles of human agents when machine agents are doing most of the work? Before an attempt is made to explore these issues, a brief review of the most fundamental legal conceptions is necessary.

Authorities and delegation

A legal philosopher, Wesley N. Hohfeld, recognized eight fundamental legal conceptions in two sets (quoted in Allen & Saxon, 1993). In each set, the paired concepts are ‘opposites’ in columns and ‘correlatives’ in rows:

Right set:	Right	Duty
	No right	Privilege
Power set:	Power	Liability
	Disability	Immunity

As discussed by Allen & Saxon (1993), all eight concepts are related to the terms 'must', 'should', 'must not' and more. Each term may involve more than one of these legal concepts. For example, 'a credit card holder must pay any outstanding amount of credit within 25 days of posting without incurring interest', implies that a card holder has the rights to pay, and not to pay until the due day, but a liability of paying interest will occur from the due date whether the amount is not paid. As pointed out by Allen & Saxon (1993), these terms may also result in multiple interpretations and it is only appropriate to expect a machine to assist the human in interpretation.

A machine agent should be seen as an assistant to the human, who can delegate some of his or her responsibilities. Only when the human user has the right and power can he or she transfer responsibilities to a machine agent. It should be noted here that what is transferred are duties or job functions rather than liability. In the same manner the government of a country can delegate responsibilities and functions to an embassy in a foreign country but cannot relinquish the liability for what the ambassador has done. Of course, this does not stop users from attempting to use software to abrogate responsibility 'sorry I can't help, the computer says so!'

To make a machine agent behave in the same way as the owner would wish, behavioural norms of its user have to be specified and stored in the machine agent. Other types of norms, such as perceptual, cognitive and evaluative, can be used to enhance the machine agent's learning ability. Finally, denotative norms will be used to design the interaction between the machine and the owner and other users. However, study of some of these norms may be more difficult because their effects are not as observable as behavioural norms.

Norm specification

Understanding of the norms and patterns of people's behaviour within an organization is a foundation for designing an effective system. In business, most rules and regulations fall into the category of behavioural norms. These norms prescribe what people must, may, and must not do, which are equivalent to three deontic operators 'is obliged', 'is permitted' and 'is prohibited'. Hence, the following format is considered suitable for specification of behavioural norms:

whenever <condition>
if <state>
then <agent>
is <deontic operator>
to <a ction> 0.

Adopting this form, a credit card company may state norms governing interest charges as follows:

whenever an amount of outstanding credit
if more than 25 days after posting

then the card holder
is obliged
to *pay the interest*
whenever an agreement for credit card is signed
if within 14 days after commencing
then the card holder
is permitted
to *cancel the agreement*.

The first norm says that after 25 days of posting the invoice, if there is still an amount of outstanding credit, the card holder will have to pay the interest. The second norm states that the card holder retains the right of cancellation of the agreement within 14 days of commencing. The next norm says that unless there is a special arrangement made, e.g. with the account manager, the card holder is not allowed to spend more than the credit limit:

whenever purchasing
if no special arrangement is made
then the card holder
is forbidden
to *exceed the credit limit*.

The card holders are expected to behave according to the norms stated in the agreements. As understood both by the customers and the credit card company, the company may impose sanctions if a customer fails to observe the norms. With this form of specification of norms, a computer program can be written to execute the norms. As long as the norms are specified, computing technologies such as active databases, object technology and artificial intelligence will have different approaches towards software realization.

Autonomy and discretion

Consider the email filtering rule we saw earlier. This rule could be restated using the form of the previous section as:

whenever a mail message arrives
if field 'Subject:' contains 'special offer'
then the mail agent
is obliged
to *move the message to Trash*.

Compare this with the sorts of rules human agents operate under. Clearly the conditions and actions of human rules do not need to be as formally stated as those of a software agent, simply because the human has more intelligence and contextual knowledge. However, perhaps the most significant difference is in the deontic condition. The email filtering rules will always be an obligation. We do not expect the filtering agent to sometimes sort our mail, when it feels

like it. No, it must always sort our mail in precisely the way prescribed. Simple email filtering is deterministic.

However, intelligent agents are expected to operate with more autonomy, no longer satisfying deterministic rules, but instead learning from experience and taking initiative. This suggests that the norms embodied in these agents should take the more general deontic form. Yet if an agent is only permitted to perform some action, why should it ever do it, and if it does not does it matter?

Human agents operate within this kind of looser deontic context. How do they cope?

Framing norms

Consider the case of a bank manager faced with an unauthorized overdraft. The published condition of the account would state (albeit in plain English!):

whenever overdraft exceed agreed limit
then the bank
is permitted
to *charge a fee*.

This is the rule which the customer has seen when taking out the account and hence legally agreed to. It thus forms a contractual norm. The customer cannot complain if the bank exercises its right to charge a fee. In practice, however, the bank manager will exercise discretion. The bank manager will not want to alienate a good customer and hence will often waive the fee.

This sounds as though a valid course of action for the bank manager would be to never charge a fee at all on unauthorized overdrafts. Clearly the customers would be quite happy with this situation, but, of course, the bank would not be! In fact, there will be corporate norms within the bank, some explicitly stated in rule books, others implicit. These will constrain the bank manager's discretion to act. One such rule might be: **whenever** overdraft exceed agreed limit

if the excess is unduly large
then the bank manager
is obliged
to *charge a fee*.

Of course, the bank will never want the bank manager to exercise discretion in an overly legalistic manner and will have further corporate norms which prevent this:

whenever overdraft exceed agreed limit
if the excess is small **and** it is a good customer
then the bank manager
is forbidden
to *charge a fee*.

The bank manager is the agent of the bank and the corporate norms allow the bank to determine the boundaries of discretionary action. Note that the contractual norm governs the behaviour of the bank with the customer. Of particular importance, the corporate norms should be such that the actions of its agent (the bank manager) ensure that the bank acts within its contractual norms.

Note that the discretion of the bank is limited by contractual norms and the discretion of the bank manager is limited by corporate norms. These are both examples of framing norms, which set bounds on the discretionary activity of an autonomous agent. Within these framing norms an agent may have its own individual norms: rules and attitudes which it uses to make individual decisions. Some bank managers are more generous than others.

Policy and meta-norms

However, even within the bounds of framing norms an agent is usually not free to have arbitrary individual norms of behaviour. Consider a bank manager who always waived the fees of family and friends, or one who waived fees of male customers but charged overdrawn women. The former would be seen by the bank as unacceptable favouritism, the latter (in the UK) would be illegal.

In fact, there are normally general policies which govern the sort of individual norms which agents are allowed to operate. In the banking context these policies would include some sense of fairness. At a legal level, UK courts use the common law concept of equity to limit the discretion of government officials and the courts themselves operate within the principle of precedent.

Notice that whereas the contractual and corporate norms we have considered operate on individual cases, all these policies operate between cases determining the sort of rule which is acceptable as a behavioural norm. These are meta-norms, norms that govern the generation of other norms.

Intelligent agents

Whereas the agents we have considered so far in email systems have been deterministic, there is increasing interest in the use of intelligent agents. These may be pre-programmed, in which case they obey norms laid out by the application developer. However, perhaps the most interesting cases are agents which use machine learning techniques to generate their own rules of behaviour. One of the early examples of this is Eager, a system which watches a user's actions, learns patterns and then when it notices a user beginning a learnt pattern offers to complete the sequence of actions (Cypher, 1991). Similar techniques have been used to select interesting news postings, build a personalized newspaper, and to generate database queries from examples of required records (Dix & Patrick, 1994).

These learning agents are also being developed for email systems. An agent can watch users' actions as they sort their mail and build automatic filtering rules: effectively capturing

the user's individual norms of behaviour. Within an organization they can even watch the patterns of document passing between individual employees and thus infer the corporate norms of behaviour and generate workflows. Furthermore, pre-programmed deterministic agents have already been proposed as part of the infrastructure of virtual organizations – it will not be long before learning agents are used at this cross-organizational level. This may include a mediation role, adapting corporate norms of one organization to those of another, rather like translators do between languages.

Such agents will start to take decisions which significantly affect the externally perceived behaviour of the individual or organization. An email agent which discards a message from the managing director could significantly affect the promotion prospects of an employee! Furthermore, it would be easy for such an agent to generate illegal rules and expose the individual or organization to litigation.

Meta-norms for agents

The autonomy and discretion of such agents must clearly be limited in the same way that human agents are. Framing rules can be specified using the deontic rules we have outlined earlier. These can be entered at an individual level or customized for the organization as a whole (for example, 'all messages from the managing director are flagged urgent'). However, perhaps the most interesting area is in the expression of meta-norms.

The learning algorithms can themselves be regarded as one level of meta-norm as they limit the sorts of rules which can be inferred. However, this is a crude and technologically determined limit. More interesting are rules which are explicitly built into the system. However, these are by their very nature far more complex than norms for case-by-case behaviour. It is always hard to step back and externalize one's norms of behaviour, how much more difficult to codify the rules which govern the generation of those norms! At the level of corporate meta-norms this may be possible as professional knowledge engineers can perform the requisite knowledge elicitation and agent programming, but it is unlikely that individual users will be able to customize their own interfaces in this way.

Meta-norms can also be captured within processes. A bank clerk who notices a more efficient way of processing loan applications cannot decide to by-pass the bank manager and approve or deny loans on their own authority. However, it may be acceptable (depending on the organizational culture) for the clerk to approach the manager and propose a rule 'we seem to always accept loans that are less than half the customers monthly salary'. Similarly, an email agent can infer filtering rules and propose them to the user using the same format as the users do to write their own filtering rules. Notice what is happening here. The software agents operate within procedural meta-norms. The agents propose new behavioural norms to the users who decide whether they fit within their own meta-norms of behaviour. The mechanism of norm generation lies with the learning agent, but authority for norm generation belongs solidly with the user.

DESIGN FOR COLLABORATIVE INFORMATION SYSTEMS

The theory of norm-based agency offers a method for designing collaborative information systems. At this point, an example of processing claims in an insurance company will be used to facilitate the discussion. This process is based on one medium-sized company in the UK (Liu & Ong, 1999). For the purpose of exposition the actual workflow has been simplified (for example, where additional information is sought in order to complete a stage). The resulting simplified workflow is typical of common practice in the insurance industry.

Workflow

Scenario-based methods are effective in modelling a series of activities. These methods are widely used in modelling human–comuter interactions (HCI) and in requirments engineering (Filippidou, 1998; Kotonya & Sommerville, 1998). One particular version (Sutcliffe, 1998) is used here to model the workflow of the insurance claim processing, which enables to identify actors, their activities and responsibilities.

Figure 4 shows the workflow of claim processing, which already uses a computer to support the business process. The computer system primarily aims at reduction of the amount of paper-flow and improvement of the document management. Nevertheless, this change has opened up a possibility of further automation to achieve a higher degree of productivity. The workflow model describes the actors and their activities in the current situation.

- An insurance subscriber, who will initiate an instance of claim processing by submitting a claim application.
- A documentation receptionist, who logs the claim on the receipt and sorts out the claims according to certain criteria.
- An officer for data entry, who digitize the claim (by performing, e.g. electronic scanning, checking and manual correction).

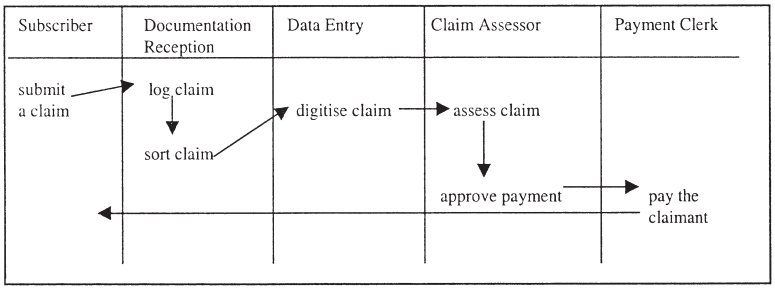

Figure 4. The workflow for claim processing.

- A claim assessor, who assesses the claim application against the case, the policy and any necessary factors, and then approves, rejects or recommends further action.
- A payment clerk, who issues the payment and an accompanying letter to the claimant.

Norm-based agency in collaborative information systems

To improve the business process further, a full-scale collaborative information system can be designed with the use of agency technology. After studying the workflow of the insurance company, it has been discovered that the whole claim processing can be divided into three major areas, covered by three groups of actors with expanded responsibility areas with support of software agents. Note that these three groups are derived from the original departmental structure, though they do not seem to match exactly. In some cases, this analysis may result in a business process re-design or re-engineering (Stamper *et al*., 1994).

Figure 5 shows a conceptual design of the collaborative information system. With reference to the earlier discussion on the norm specification and the levels of embedded norms (see Figure 3), norms at different levels have different scope of impact and characteristics.

Within each group, for example group 1, there will be norms to prescribe how a claim should be logged, and to specify a standard way of digitising a claim and criteria for sorting. In the group, there are a manager, supervisors, officers and clerks with the support from software agents. The norms at this level are behavioural in their characteristics (often prescribing actions) and can be explicitly specified. These can therefore be embedded into software agents to automate some of the business functions. For example, the first norm below is embedded into an agent. When the assessor identifies and informs the agent that one receipt is not original but a copy, the agent can check the amount and act according. The agent

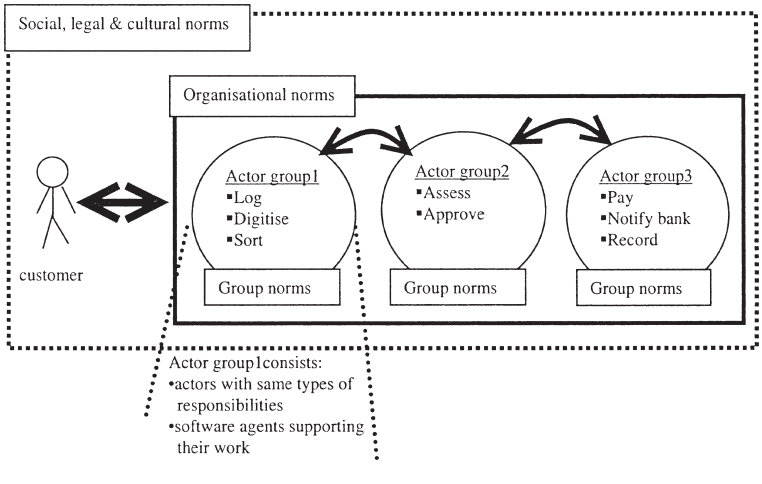

Figure 5. Norms at different levels in collaborative information systems.

prompts the assessor that he or she can reject the claim if the amount is more a threshold, e.g. £200. The second norm states that as soon as a claim is approved, the account payable department must pay the amount to claimant within 14 days.

whenever a claim is not accompanied with original authenticated evidence
if the amount of claim is more than £200
then assessor
is permitted
to reject the claim
whenever a claim is approved
then account payable
is obliged
to issue the payment within 14 days.

The norms at the group level are local to a specific domain or an actor group. On the other hand, norms at the company level, labelled as 'organizational norms', apply to the whole insurance company. They describe company-wide policies and ground rules for handling customers and claims.

Most of the norms we have discussed are local – prohibiting or requiring specific individual actions. Organizational norms may be of this form; for example, requiring that only indelible ink is used on certain documents. However, at this level there are often also broader norms, which impact on the entire work process. For example, the company policy states that claims are expedited as quickly, efficiently and equitably as possible. Such broad corporate norms must be 'implemented' by the individual norms. That is, the emergent behaviour resulting from the individual norms should ensure the satisfaction (or at least not inhibit) the company's end-to-end process norms.

One approach to workflow management is to attempt to ensure this compliance by formalizing the business processes within an automated workflow system, that is codifying the process norms. Although this works in certain corporate environments, many workplace studies have found that complex work processes are typically carried out within a richer ecological setting (e.g. Flor & Hutchins, 1991; Rouncefield *et al*., 1994; Sellen & Harper, 1997). These studies and our own previous work suggest that prescriptive attempts to codify work practices run the risk of destroying the very processes they hope to capture. In contrast, a descriptive codifying of the norms which contribute to a work process can allow us to assess the robustness of the process.

Finally, there is another level, the social, legal and cultural context where the customer conducts business with the company. Norms at this level have to be observed by the customer and the company as the basis for trust and all business activities. Some norms are based on accepted mores whereas others are derived from legal grounds, or both. For example, in submitting a claim, one is expected to present a complete truthful account of the incident. Both the customer and insurance company understand that any fraudulent claims, which are illegal and immoral, may result in legal penalties. Some of these norms are embodied in criminal law (e.g. fraud) or civil legal contracts. However, the fundamental mores and legal principles

can be seen as 'meta-norms', as discussed in above. They provide a general framework for individual operational norms – they are the norms which govern the generation of norms at other levels.

CONCLUSIONS

Organizations are information systems where people use signs to conduct business activities. Human agents act in the systems in an organized and co-ordinated manner. In achieving this organization and co-ordination, norms play an essential role in governing people's behaviour. To understand and model the behaviour of members of an organization becomes essentially the task of understanding and representing norms. There are many types of norms which are rooted in rich social, cultural and linguistic contexts. The norms govern a whole range of human activities from perception, cognition, and evaluation to action; and they can be explicit as well as implicit. Therefore, modelling norms can be a difficult task and can only be achieved to a limited extent.

Norms are closely related to legal concepts such as responsibility and authority. Framing norms are concerned with delineation of such boundaries, within which human agents have the right to make decisions and to exercise discretion. When machine agents are employed in collaborative information systems, some responsibilities will be delegated to machines. The delegation is only valid within the boundary of users' authority.

In illustrating our discussion, email systems have been used as a driving example. Norms can be embodied into email systems at various levels: individual norms, group norms, corporate norms or social norms. At present these are normally of a deterministic nature. However, the norms within which human agents act are of a richer nature. To capture these we have incorporated deontic operators into explicit rules. Human agents operate within the limits of framing norms (including corporate norms and contractual norms). Their discretion within these limits is governed by meta-norms (equity, fairness, etc.).

When this method of norm-based agency is applied in designing of a collaborative information system, it provides an effective approach to analysing and modelling the business processes where multiple actors are involved. It then guides the analysis into identifying and grouping actors who share common responsibilities. On the basis of identified grouping, norms governing patterns of behaviour at different levels can be identified and some of them can be explicitly specified as the basis for programming the software agents. The software agents can then perform delegated responsibilities and support the work processes. This approach has addressed and also provided some answers to many important philosophical and methodological questions in collaborative information systems for enterprise decision support, e-commerce and other applications.

ACKNOWLEDGEMENTS

A part of this work was presented at the workshop on Computational Semiotics, at Leonard De Vinci University, in Paris 1997. This work is partly supported by the UK/HK Joint Research Scheme project 9050101.

REFERENCES

Allen, L.E. & Saxon, C.S. (1993) A-Hohfeld: A language for robust structural Representation knowledge in the legal domain to build interpretation-assistance expert systems. In: *Deontic Logic in Computer Science*, Meyer, H.C. & Wieringa, R.J. (eds). Chichester, John Wiley & Sons.

Clark, H.H. & Brennan, S.E. (1991) Grounding in communication. In: *Socially Shared Cognition*, Resnick, L.B., Levine, J. & Behreno, S.D. (eds). Washington DC, American Psychological Association.

Cypher, A. (1991) Eager: programming repetitive tasks by example. In: *Proceedings of CHI'91*, pp. 30–39. New Orleans, ACM Press.

Dix, A. (1994) Computer-supported cooperative work – a framework. In: *Design Issues in CSCW*, Rosenburg, D. & Hutchison, C. (eds), pp. 9–26. Lancaster, Springer Verlag.

Dix, A. & Patrick, A. (1994) Query by browsing. In: *Proceedings of IDS'94: the 2nd International Workshop on User Interfaces to Databases*, pp. 236–248. Lancaster, Springer Verlag.

Dix, A., Finlay, J. & Hassell, J. (1994) Environments for cooperating agents: Designing the interface as medium. In: *CSCW and Artificial Intelligence*, Connolly, J. & Edmonds, E. (eds), pp. 23–37. Lancaster, Springer Verlag.

Eichmann, D. (1994) Ethical web agents. In: *Electronic Proceedings of the Second World Wide Web Conference,* http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Agents.

Filippidou, D. (1998) Designing with scenarios: a critical review of Current Research and Practice. *Requirements Engineering*, **3**, 1–22.

Flor, N.V. & Hutchins, E.L. (1991) Analysing distributed cognition in software teams: a case study of team programming during perfective software maintenance. In: *Empirical Studies of Programmers: Fourth Workshop*, pp. 36–64.

Gaver, W. (1992) The affordances of media spaces for collaboration. In: *CSCW'92*. Toronto, ACM Press.

Holt, A.W. (1985) Coordination technology and Petri Nets. In: *Advances in Petri Nets*, Rozenberg, G. (ed.), pp. 222, 278–296. Berlin, Springer-Verlag.

Klein, M. (1999) The multi-agent approach to enterprise information systems: challenges and directions. In: *ICEIS 1999 (International Conference of Enterprise Information Systems)*, Setubal.

Kotonya, G. & Sommerville, I. (1998) *Requirements Engineering: Processes and Techniques*. Chichester, John Wiley & Sons.

Kreifelts, T., Hinrichs, E., Klein, K.H., Seuffert, P. & Woetzel, G. (1991) Experiences with the DOMINO office procedure yystem. In: *Proceedings of ECSCW'91*, Bannon, L., Robinson, M. & Schmidt K. (eds), pp. 117–130. Amsterdam, Kluwer.

Krogh, C. (1995) The Right of Agents. In: *Intelligent Agents II: Agent Theories, Architectures and Languages*, Wooldridge, M., Muller, J.P. & Tambe, M. (eds), pp. 1–16. Berlin, Springer.

Liu, K. (2000) *Semiotics in Information Systems Engineering*. Cambridge, Cambridge University Press.

Liu, K. & Ong, T. (1999) A modelling approach for handling business rules and exceptions. *The Computer Journal*, **42**, 221–231.

Malone, T.W., Grant, K.R., Lai, K., Rao, R. & Rosenblitt, D. (1987) Semistructured messages are surprisingly useful for computer supported coordination. *ACM Transactions on Office Information Systems*, **5**, 115–131.

Nwana, H.S. & Ndumu, D.T. (1996) An introduction to agent technology. *BT Technology Journal*, **14**, 55–67.

Prinz, W. & Kolvenbach, S. (1996) Support for workflows in a ministerial environment. In: *CSCW'96*, pp. 199–208. Lancaster, ACM Press.

Rao, A.S. (1995) *Decision Procedures for Propositional Linear-Time Belief-Desire-Intention Logics.* In: *Intelligent*

Agents II, Wooldridge, M., Muller, J.P. & Tambe, M. (eds), pp. 33–48. Springer, Berlin.

Rouncefield, M., Hughes, J.A., Rodden, T. & Viller, S. (1994) Working with 'constant interruption': CSCW and the small office. In: *Proceedings of CSCW'94*, Furuta, R. & Neuwirth, C. (eds), pp. 275–286. Chapel Hill, NC, ACM Press.

Searle, J.R. (1969) *Speech Acts – an Essay in the Philosophy of Language*. Cambridge, Cambridge University Press.

Schmidt, K. & Simone, C. (1996) Coordination mechanisms: towards a conceptual foundation of CSCW systems design. *Journal of Collaborative Computing*, **5**, 155–200.

Sellen, A. & Harper, R. (1997) Paper as an analytic resource for the design of new technologies. In: *Proceedings of the 1997 Conference on Human Factors in Computing Systems, CHI' 97,* pp. 319–326.

Smith, R. & Mamdani, E.H. (1996) Intelligent software technologies. *BT Technology Journal*, **14**, 24–31.

Stamper, R. & Liu, K. (1998) Collaborating through shared meanings, social norms and distributed authority, In: *Readings in CSCW*, Vertegaal, R. (ed.), pp. 21–38. Enschede, CTIT.

Stamper, R., Liu, K. & Huang, K. (1994) Organisational morphology in re-engineering. In: *The Second European Conference on Information Systems*, pp. 729–737. Nijenrode, Nijenrode University.

Stamper, R., Liu, K., Hafkamp, M. & Ades, Y. (2000) Understanding the role of signs and norms in organisations, – a semiotic approach to information systems design. *Journal of Behaviour and Information Technology*, **19**, 15–27.

Sutcliffe, A. (1998) Scenario-base requirements analysis. *Requirements Engineering*, **3**, 48–65.

Teege, G. (2000) Users as composers: part and features as a basis for tailorability in CSCW systems. *Journal of Collaborative Computing*, **9**, 101–122.

Winograd, T. & Flores, F. (1986) *Understanding Computers and Cognition: A New Foundation for Design*. New York, Addison-Wesley.

Wooldridge, M. & Jennings, N. (1995) Intelligent agents: theory and practice. *The Knowledge Engineering Review*, **10**, 115–152.

Wright, G.H.V. (1963) *Norms and Action – a Logical Enquiry*. New York, Routledge and Kegan Paul.

Biographies

Kecheng Liu holds a chair professor position at Staffordshire University and is director of University's research centre for e-business systems. His research interests span from requirements studies, systems analysis and design, object methods for systems engineering, normative modelling of software agents, to HCI and CSCW. He is a leading researcher in Organizational Semiotics, where one single authored and two jointly edited books by him can be found.

Lily Sun with a PhD in information requirements for strategic decision support, is Senior Lecturer at Staffordshire University. Her research interests and recent publications are found in information systems, requirements engineering, IT for strategic management and e-commerce systems.

Alan Dix is a mathematician by training (and still deep in his heart) and has worked in academic and commercial computing since 1981, researching particularly in the areas of CSCW and HCI, where his co-authored textbook is one of the leading international texts. He is currently Professor of Computing at Lancaster University, UK and director of two Internet companies: vfridge (http://www.vfridge.com) and aQtive (http://www.aqtive.com).

Murali Mohan Narasipuram is Associate Professor at City University of Hong Kong since 1990, after working in the software industry in India 1982–90. Areas of his academic interests include Electronic Commerce, Artificial Intelligence, Information Systems Development and Methodologies.