

**A social actor understanding of the institutional structures at play
in information systems development**

Author

Rowlands, Bruce

Published

2009

Journal Title

Information Technology & People

DOI

[10.1108/09593840910937490](https://doi.org/10.1108/09593840910937490)

Rights statement

© 2009 Emerald. This is the author-manuscript version of this paper. Reproduced in accordance with the copyright policy of the publisher. Please refer to the journal's website for access to the definitive, published version.

Downloaded from

<http://hdl.handle.net/10072/28502>

Griffith Research Online

<https://research-repository.griffith.edu.au>

A social actor understanding of the institutional structures at play in information systems development

Bruce Rowlands

School of ICT, Griffith University, Brisbane, Australia

Abstract

Purpose — to provide insights into the process of applying Lamb & Kling's (2003) *user as social actor model* as an institutional lens to analyse data and to generate findings of IS practice.

Approach — provides an example from a recently published research project (Rowlands, 2008) why the *social actor model* was chosen, and an application of how the model was operationalised in data collection and analysis.

Findings — provide two illustrations of theorising *the social actor model* at different levels: at an elementary level by using instantiation to map the model's constructs to a match in the transcript text; and at a higher level where the model enabled the author to tie together a number of individual phenomena to construct a broader higher-level schema.

Originality/value — as the *social actor model* is relatively new, and few guidelines or examples exist, the paper provides rich insights into the difficulties with the application of this model and a solution for coping with these so that other researchers can read of an example that may be similar to their own and therefore guide their work.

Keywords User as social actor model, institutional theory, information technology (IT) enactment.

Paper type Reflective Essay

INTRODUCTION

This paper provides insights into the process of applying Lamb & Kling's (2003) *user as social actor model* to generate findings about information systems (IS) practice. The practice under study was of how systems developers enacted a systems development methodology (SDM) in their day-to-day work (Rowlands, 2008). The source of data comprised thirty interviews with systems developers from a large IT department within a major Australian bank.

The focus is on describing why I chose the social actor model, my understanding of the model, the research process, theorising at various levels and the research methods I used rather than the findings. This format is important as the *user as social actor model* is relatively new, and Lamb & Kling provided few methodological guidelines for operationalising the four dimensions and sixteen constructs central to theorising about the enactment of information & communication technology (ICT) in practice. I claim that this format is instructive and innovative in the sense that there is theoretical value in drawing on and incorporating contributions from a prior case study outside its original domain.

Reasons why I chose the Social Actor Model

It is important to provide background to how I initially 'came across' the *social actor model*, and how I understood and now understand the concept of *the user as a social actor*. First of all, I became acquainted with *social actor model* in 2004/5 at about the same time as I was undertaking a major field study into how systems developers enacted their SDM in practice. What caught my attention was that the model was created using a style of grounded theorising — a research method that I had a working knowledge of, and interest in. However, the *social actor model* did not inform my data collection and nor was I developing a grounded theory. The topic of methodology enactment had been studied previously, but I thought it was not well understood with most studies focussing on the technical aspects of the SDM, or only on developers as the unit of analysis. Because I believe that IT practice should be seen as more than a technical activity and that SDMs are organizationally embedded artefacts, I was searching in my mind for

To be published in *Information Technology & People*, 2009, 22(1)

an appropriate theory and analytical framework that addressed issues of the technological artefact (the SDM), the role that developers play in enacting the technology, and included different levels of analysis. On first reading *Reconceptualising Users as Social Actors in Information Systems Research* (Lamb & Kling, 2003) I was struck by the paper's complexity (length, writing style) and novelty in terms of challenging the *user* concept. Yet, I was pleasantly surprised that I could 'get my head around' a key finding of their study — a model presented as a two dimensional table — comprising four dimensions of a social actor and 16 constructs of connected and situated individuals.

A defining point about the model that I found appealing for the analysis of my field data was that the model, via the four dimensions, operates at multiple levels. According to the *social actor model* the dimensions *interactions* and *identities* relate organisationally situated individuals to others and to the technologies they use to interact with and present themselves to others. The second two dimensions — *affiliations* and *environments* relate people to their organisation, and to the industries and environments of those organisations. It was this multi-level focus that instantly attracted me to the model, as I knew from questioning and listening to the interviewees in the field study, that their use of the methodology was influenced by factors both in and outside of the organisation, some were unique and individual; others were related to the project or the organisation as a whole.

By being able to use the model at a work practice level, I favourably contrasted the simplicity and power of this model in comparison to structuration theory, which I classify as a meta theory operating at a much higher level of generality. I always considered structuration theory more of a view from high to the exclusion of understanding individual motivations, and likewise local issues as a context for action. On the other-hand, the *social actor model* (supported by institutional theory), can be a model of practice operating at a lower level of generality than structuration theory, with an emphasis on analysing the organisational setting in terms of developer's *interactions* with organisational *affiliates* and the *environment*; and most importantly the impact of organisational constraint on *individuals*.

Another key insight that took me a little longer to appreciate and understand was the notion of a *social actor* as a unit of analysis. I now understand from Lamb's (2006) later writing that the term *social actor* is a construct that conflates people's interactions (giving meaning to the term "social"), their information environments (taking into account context), and their technologies. According to the *social actor model*, the user is not a technically focused or socially thin, passive user of technology, but a person who acts with information technology in a social setting. To avoid potential confusion, I need to point out that in this example involving systems developers, it is the developer who is the 'user' of the technology (the SDM), in contrast to conventional IS literature, where the user is often portrayed (let's say) as an office worker being the recipient of a developed system.

In sum, *the user as social actor model* was chosen for pragmatic and methodological reasons. The model was new, celebrated, and I saw it enthusiastically as an opportunity to 'try it out'. Lamb & Kling's (2003) paper was awarded the 2003 *MIS Quarterly* best research paper award. Second, as Lamb & Kling (2003: 219) offered, "the model provides a framework for the systematic research of complex, highly contextualised ICT use in organisations, rather than the study of isolated aspects of ICT use in decontextualised settings". Third, I wanted to study the systems developer at the micro, meso, and macro level. Fourth, the four dimensions and sixteen characteristics of the model seemed understandable, and relatively easy to operationalise in my research design. Later I was to realise that operationalisation wasn't that simple, but the choice of the model was fortuitous because of its focus on understanding how institutional structures and context shape IS related practice.

THE RESEARCHER'S THEORETICAL LENS

In 2005, I began reading the work of Kling and colleagues (Kling et al, 2005) to help my understanding of the foundations of the *social actor model* and therefore apply it authentically, as the authors would have intended. I went on to read that people's individual autonomy (their agency) and their work practices are shaped by the social norms (dress standards, working

hours), routine work practices, and other social and physical structures that surround them (such as office layouts, and technical artefacts). By agency, I mean the extent to which individuals – in this case developers – have the ability to shape their work practice, as opposed to the extent to which their working life is structured in ways out of their control. Importantly, in terms of the research example in the bank, I deemed structure to include work procedures mandated by the SDM such as gaining sign-off before commencing the next stage of development, the day-to-day routine interactions within and among project groups, and authority relations based on power, occupational hierarchy, and expertise among the bank's IT department. On reading the transcripts, it became apparent to me that systems developers can be seen as complex social actors acting in constrained ways, rather than simple "users" of the SDM. Given this view, I also formed the opinion that the SDM operates as an institution involving various structures around which the systems developers operate.

From Scott (2001) I understand institutions to be social structures based on taken for granted formal or informal rules that exert influence and regulation over other social entities as a persistent feature of social life. Using this definition, I consider the SDM has become institutionalised within the bank because of its longevity of use and its associated practices having become routine. Similarly, as the transcripts show, the SDM goes relatively unnoticed where developers take it for granted in performing their day-to-day work.

Institutional Theory

Drawing on the work of Scott (2001), Lamb & Kling describe how the social actor concept has been theoretically supported by new institutionalist approaches, whereby institutions provide a framing context within which social actors make constrained choices about technology use, particularly when they are situated in organisations. According to Scott (2001), new institutional theory is really a body of knowledge that studies the relationships between organisations and their environments focussing on how structures become established (or institutionalised) as guidelines for social behaviour. A central principal of this theory is that institutions are transmitted by various types of carriers, including technical artefacts (Scott, 2001: 81) such as a systems development methodology that guides IT practice.

According to my understanding of new institutional theory, the term 'enactment' means more than 'use'. 'Use' is a simple word conjuring up an image of the SDM as an objective tool, and implying a separation between the technical and the social (the context of its use). Enactment therefore is a process in which social actors (systems developers) respond in a dynamic interplay between order within the bank (structure), their individual freedom (agency), and how the SDM is perceived by developers to build systems in a specific project situation. Most importantly, this interplay is situated in a social and historical context and is bounded by physical surroundings and technological artifacts including the SDM. Given this working definition of enactment, the example case applied the *user as social actor model* as a conceptual lens to understand methodology enactment within a single organisational setting.

Social Actor Model

Lamb & Kling's (2003) framework focuses on the *use* of ICTs and aims to develop a social actor model of 'users' in an organisational setting. My problem was in deciding whether I could legitimately transfer the framework from one setting (*use*) to another (*development*) substituting SDM for ICT. In arriving at a solution so I could apply the model authentically, I had to be satisfied of two fundamental conditions: that I could define SDM as a technology and that there were sufficient similarities in work practice involving both technologies. I answered 'yes' to both conditions after some deliberation.

Firstly, I define a SDM as an organisational technology. This definition aggregates task, technique, knowledge, tools and process into a single construct – *technology*, recognising that there is more to the term than just hardware.

Second, I had to think deeper about equating ICT use as having similarities with systems development. My line of thinking at the time went something like this. There are several parallels between ICTs and SDMs: they are both artefacts that can (and do) regulate the work of their

To be published in *Information Technology & People*, 2009, 22(1)

respective 'users', and they are both configurable in nature. By configurable, I mean that in their design and use, they are interpretively flexible and adaptive in use. There are also many similarities in the work situation. For example, in modern organisations, 'users' of technology often circumvent their job tasks through a process known as technology appropriation. Similarly, developers can build systems without paying too much attention to all aspects of the SDM. Furthermore, in my research, systems developers enacted the SDM to build a product or a service – documentation or systems. Users in the original domain (Lamb & Kling, 2003:230) came from the pharmaceutical, law, and real estate industries where the work practice involved professionals using ICTs to construct services and products online. I concluded that given these similarities in technology and work practice I could legitimately transfer the framework from ICT use to SDM enactment demonstrating an application of the *social actor model* outside its original domain.

My research made use of Lamb & Kling's (2003) *social actor model* by illuminating methodology enactment at four levels: individual, project, organisational, and industry. The following paragraphs provide description of the model's four dimensions in terms of the context of the research example.

Affiliations represent inter- and intra-organisational relationships created and supported by organisational members as a result of their day-to-day activities. Systems developers work together with the methodology comprising professional networks. These networks exist within the bank but also apply to the IT and financial industries as well, and to a wider national and international context.

Environments. The environment an organisation operates in is shaped by the kind of affiliations it has formed with (for example) technology suppliers, customers, government agencies, the banking and technology media, and other organisations that produce similar services or products, known collectively as an *organisational field* (Scott: 2001). Methodology enactment needs to recognise the normative, regulatory, and cognitive institutionalised practices of the bank, and other associations that circumscribe organisational action.

Interactions. Systems developers see themselves as organisational members working with others (project members, managers, and the client) enacting the methodology (and other media such as email) in support of their interactions. Information and resources are mobilised as systems developers socially engage internally, and with *affiliated* organisations.

Identities. Systems developers regularly enact SDMs to compile and present information to various affiliates. In so doing, they create an identity for their organisation and for themselves. Systems developers are defined by their presentations as individuals (analyst programmer) or a collective entity (systems developer).

Each of the four dimensions is comprised of four further characteristics or behaviours of connected and situated individuals. These sixteen behaviours were used to develop the initial coding scheme for the qualitative analysis of data to be discussed next. Only the *Interactions* (c.f. Table 1) has been included for illustration purposes as space limitations preclude the inclusion of the remaining three tables for the *Affiliations*, *Environment* and *Identity* dimensions.

METHOD ISSUES

Research Approach

The research centred on exploring the enactment process and how systems developers carry out their work in the bank. Thus, essentially concerned with organisational rather than technical issues, I determined that the case study was highly appropriate for this purpose. The banking and financial services sector was deemed appropriate because of the extremely important role that ICTs play in the success of companies in this industry, and importantly, the banking industry is highly technical, highly competitive, highly regulated and institutionalised.

The unit of analysis for the case study comprised *individuals*: systems developers (including analysts, programmers, project leaders, project managers, and the CIO); a variety of new development and maintenance projects and the *interactions* and *affiliations* within these projects;

the internal culture and norms of the bank – the organisation; and the external business and industry *environment*. The analysis was therefore multi-level.

According to Lamb & Kling (2003:221), *the social actor model* can be used to guide data collection and to pose interview questions. I didn't do that. Instead, the model was used only in data analysis as a theoretical lens to draw meaning from the interview data. It helped make sense of what occurred in the field, provided a set of constructs to be investigated from the interviews, and guided my interpretation and focus.

In the next section, I provide two illustrations of theorising *the social actor model* at different levels: at an elementary level by using instantiation to map the model's constructs to a match in the transcript text; and at a higher level where the model enabled me to tie together a number of individual phenomena to construct a broader higher-level schema. One schema was a focus on the power relations between systems developers and the business client at the workplace.

Data Analysis

The analysis of transcripts involved five steps and follows an identical approach to analysis first reported in Rowlands (2006) and subsequently used in Rowlands (2008), and Van Akkeren & Rowlands (2007).

Step 1: *Interpretive translation of social actor model*. In this first step, the sixteen characteristics and behaviours of connected and situated individuals (Lamb & Kling, 2003:213) were interpreted by myself and translated to an equivalent meaning in terms of the case scenario. For instance, in the dimension *Interactions*, the first behaviour is that "*organisational members seek to communicate in legitimate ways*" (c.f. Table 1, row 1, column 2). I had already conducted the interviews and had re-read the transcripts a number of times. By being familiar with the context, and through a process of understanding from a source domain (the familiar – the sixteen social actor constructs) to a target domain (the relative unfamiliar – the interview text), I interpreted or instantiated the first behaviour to mean "*the SDM mandates documentation throughout all phases & calls for meetings, both formal & informal among affiliates to review them*" (c.f. Table 1, row 1, column 3). While multiple interpretations were possible, and through a process of iteration and revision, I settled on an interpretation taken from the interview transcripts that best fitted what I thought was the meaning of the original construct. This process of interpretation and translation (or mapping) of the original constructs to a match in the transcripts continued for the entire sixteen behaviours of the connected and situated individual. For me, this elementary form of theorising through use of interpretation provided both a way of thinking and making sense about actions of actors in the organisation; and a way of seeing the data in analysis. However, at the time of conducting step 1, I was not fully cognisant of the power of interpretation as a theorising tool. While I appreciated its usefulness in operationalisation, it is only now that I can claim that interpretation was driving my basic understanding and theorising about the social setting.

Step 2: *Assigning codes*. Based on key words from the *user as social actor model*, codes were assigned to match the examples from the case (c.f. Table 1, column 1). I created and operationalised a list of codes prior to content analysis (step 3) based on the model's framework. By operationalisation, I mean that the concepts from *the social actor model* were expressed in coded elements that could be identified in textual analysis.

Step 3: *Content Analysis*. In attempting to gain theoretical understanding of the complexity of the text, the *social actor model* was used in a form of content analysis where the text was systematically listed, coded and categorised according to the sixteen behaviours of the connected and situated individual (described in column 1 of Table 1). The list of researcher-constructed labels that best captured the description of the phenomenon was then deductively applied to the text to codify, count, and extract the data associated with each interview. This same format was carried through the entire thirty interviews.

Step 4: *Managing the Data*. In terms of data management issues, the process of analysis was assisted by and recorded in a database through procedures such as importing transcribed interviews, coding against the sixteen constructs, adding comments and reflections, sorting the

interpretations by code; and text retrieval of selected instances into the body of the research report.

Step 5: *Writing the narrative*. The last step in analysis involved presenting a “story” piecing together interviewee accounts taken from the entire case study database, guided by the theoretical lens offered by the *social actor model*. In this narrative (with sample excerpts shown in the following section), I presented informed perspectives in a coherent and convincing story. To assist in this ‘understanding’, the *social actor model* provided a focus on specific social, organisational, and individual phenomena in a particular institutional setting – the workplace. The model enabled me to tie together a number of individual phenomena to construct a broader higher-level schema. One schema (as reported in Rowlands, 2008) was a focus on the power relations between systems developers and the business client at the workplace. While step 1 described theorising (using instantiation) at an elementary level, step 5 involved theorising at a higher level of abstraction where I moved away from directly observable individual and organisational activities towards theorising at a level of abstraction higher than the empirical world involving non-observable, structural issues such as power. It was at this level that I began to focus on organisational structures and constraints and their influence on the systems developers.

FIELD TO DATA TO CONCLUSIONS

In terms of theoretical contribution, the case aimed to produce an understanding of the social and institutional context of SDM enactment. By way of empirically supported examples, the excerpts below illustrate the constraining aspects of the SDM, and how the SDM manifests itself as *institution* in day-to-day actions within the bank.

Interactions

Social Actor Dimensions and Code	Characteristics & Behaviours of Connected and Situated Individuals (Lamb & Kling, 2003:213)	Examples from the Case
Interactions [IN]		
[IN-DOCN]	Organisational members seek to communicate in legitimate ways	The SDM mandates documentation throughout all phases & calls for meetings, both formal & informal among affiliates to review them.
[IN-ACTION]	Organisational members build, design and develop interactions that make information actionable	The SDM mandates the generation of specifications becoming actionable documents requiring a sign-off at each stage. This is the work culture imposed by business clients on IT.
[IN-TAILOR]	ICTs become part of the interaction process as people transform, tailor and embed available informational resources into connections and interactions	Developers tailor the SDM in pragmatic and ingenious ways as part of the design documentation process.
[IN-CONSTR]	As organisational members, people perform socially embedded, highly specialised actions on behalf of the organisation	Using the SDM dictates and constrains their role within the bank – it tells developers what they must do – it’s the bank’s way.

Table 1. A Social Actor View of *Interactions*

According to the *social actor model* individuals are involved in networks that take shape within and among organisations. Networking refers to the *interactions* where organisational members work and interact with others (affiliates) using a methodology (and other media) in support of their interactions. In terms of the bank, I saw the SDM as a vehicle to bring together project members to coordinate their tasks when interacting with clients, industry bodies and business partners. The *interaction* dimension was by far the most prominent in the interviews as it describes systems developers in their day-to-day work role networking and relating themselves to others. My analysis identified a range of interaction behaviours such as producing documentation, seeking sign-off, tailoring or circumventing their job tasks, and acting in constrained ways.

To be published in *Information Technology & People*, 2009, 22(1)

According to *the social actor model* in Table 1, organisational members seek to communicate in legitimate ways. I sought examples of this, and in its most visible form, the SDM mandates documentation [IN-DOCN] throughout all phases of the development lifecycle and requires project leaders to call for meetings, both formal and informal among affiliates to review project status and sign-off on documentation. It is the requirement to attend meetings and the necessity to gain approval - both forms of legitimate communication that are governed by the SDM. We see here an example of the SDM (as institution) regularising the work of developers. Below is an example of a programmer citing the amount of documentation required as a lot of red tape and excessive time:

basically the main argument against this red tape or what is seen as a red tape is the time factor involved. ...obviously the more approvals you need the more people you need to contact. If certain people aren't around then it takes longer for the approval to come through.

A further dimension of the *social actor model* was that while organisations are connected as networks, within those networks, relationships with other actors are often in misalignment or even in conflict. This condition puts pressure on organisation members to develop interactions that present information in an acceptable, standardised format. *Social actor model* asserts that when this information is exchanged in this format it is made actionable. In terms of the case example, I translated this as the methodology mandating the generation of specifications that become actionable documents [IN-ACTION] requiring a sign-off at each stage. Developers were forced to do things. I provided evidence that developers viewed sign-off in two ways: enabling – to gain approval so that work could commence on the next stage; and constraining – as a way whereby business clients maintain power and authority over the development process. A project manager mentioned a structural role of the methodology, through sign-off, it enabled the business client to keep financial control of the project:

...you have to get sign off at various points. Yes, the methodology is used by the technology people to build things. But, before you can get funding for the next stage the technology group needs to provide to business things for the project to then proceed to the next phase. So if you want funding to go on further, you'll need to do things. So it [the SDM] forces you to do things.

Further evidence, and another prominent interaction code among the transcripts was that using the SDM and following routine patterns of work dictated and constrained [IN-CONSTR] their role within the bank – it told developers what they must do – it's the bank's way thereby enforcing the regulations and the authority structures of systems development, etc. I provided examples such as the following where one senior analyst believed that business used the methodology as a form of control by locking development teams into unreasonable schedules making it very difficult for the developers and in the end, producing poor quality systems because:

.. it means that your development time is locked in. Usually, your technical needs are bolted. You're told that you've got to develop a system, so there always has to be an estimate. So what it means to me is that once you've signed-off on a certain amount of money those requirements will be delivered by the end of the project. We're sort of locked in.

And, the upshot of this constraining work practice, according to the same analyst was that in a typical project:

we've got this time frame, we have to deliver this project. What you typically do is try and develop 100% of it but what you end up doing is developing 100% of it at 80% quality, instead of developing 80% of it at 100% quality. So you have this thrashing period in the last months of the project where developers are working long hours and making mistakes, causing errors which cause more problems, and so it goes on.

A further prominent *interaction* identified in Table 1 that served to influence the enactment of the SDM was that as part of the interaction process between networks, organisational members use their professional knowledge to manipulate (tailor and transform) the SDM in order to complete the task at hand, devising different ways of enacting the SDM in the process [IN-TAILOR]. Many developers mentioned that they did not follow the method slavishly and that it

To be published in *Information Technology & People*, 2009, 22(1)

was often circumvented because some required tasks are simply overkill. Similarly, depending on the size of the project, developers didn't want to be doing everything to the same level of detail all of the time. Time was the main reason for tailoring or circumventing the methodology, as a junior analyst described:

If a small project only has a few parties involved and those few people already know each other and the task that we are taking is routine, then that doesn't require much documentation or formal sign off. In that sense taking time to do something that isn't really necessary is a waste of time.

The above example provides evidence that developers do have active agency, that is, some ability to shape their working practice although as the excerpt illustrates, the constraints of their work environment means that developers are often limited in what they can do. The degree to which structure and agency are allowed depends on the context.

I concluded my analysis, by saying that by taking an institutional perspective with a specific focus on structures at the individual, project, organisational and environment levels, the above examples provided a grounded description showing how the power relations structurally embedded within the SDM influenced enactment within the bank. This institutional perspective, through the specific frame of the four dimensions, had not been applied to SDM enactment and hence provides a fresh perspective on contextual issues surrounding SDM enactment than reported on before. The findings show how systems developers' concerns and everyday work is shaped by stabilised work practices and the unequal authority relations that exist between the business client and systems developers. Another key finding is support for the proposition that SDMs are more than just technical artefacts but are institutions that exert their own type of agency, and that SDMs also interact with human agency in the systems development process.

DISCUSSION AND CONCLUSION

In reference to the utility of the concept of the *user as social actor model*, I claim that through the interpretive translation of the model, the sixteen coded behaviours drew a multi-level focus on the way the SDM was enacted, that the case provides evidence for Lamb & Kling's (2003) central argument that the systems developer is best conceptualised as a social actor and not a technically focussed methodology user; and that there is an explicit link between individual actions and a larger social context. This claim, of providing further evidence is important in the social sciences. For instance, research such as this where the model is extrapolated outside its original domain demonstrates the model's explanatory power by transcending the unit of analysis to include the individual, workplace and industry level.

All research designs have limitations. In Rowlands (2008) I wrote that "the findings are not suggesting that we can use the case as a 'test' of the model". While my opinion still holds, I began to question this assumption as a number of reviewers from tier A journals had written complimentary remarks about the adequacy of my work in terms of evaluation and testing of the *social actor model*. This contradiction between my beliefs and the comments of reviewers led me to re-visit the writings on interpretive epistemology. I concluded from my readings a key reminding point: observed data are always pre-theorised, and that the interview text I gathered and interpreted about work practices and institutional concepts was understood through my particular 'way of seeing'. Analysis, through the coding of transcripts, was in other-words, theoretically informed by the model. In sum, the social actor model (the conceptual lens) and methodology enactment (the phenomena being theorised) are inter-dependent. Hence, 'social theory' cannot be tested or 'filled out' with 'data' whose meaning is taken to be objective or independent of researchers. For me, this was an important reminder that qualitative research is value-laden.

In conclusion then, rather than using the case as a 'test' of the model, I wrote in Rowlands (2008) the case succeeds in establishing the plausibility of the *social actor model* by providing evidence of its capacity through concept instantiation to provide meaningful analytical codes; and as argued provides a fresh perspective on contextual issues surrounding SDM enactment than reported on before. Lastly, I concur with Lamb and Kling (2003) that their contextual model offers relevance in helping IS researchers to better understand ICT related phenomena.

REFERENCES

- Kling, R. Rosenbaum, H. and Sawyer, S. (2005) *Understanding and Communication Social Informatics: a framework for study and teaching the human contexts of ICTs*, Information Today, Medford NJ.
- Lamb R (2006) "Alternative Paths Toward a Social Actor Concept", *Proceedings of the Twelfth Americas Conference on Information Systems*, Acapulco, Mexico, pp. 4113-4123.
- Lamb, R. and Kling, R. (2003) "Reconceptualising Users as Social Actors in Information Systems Research", *MIS Quarterly*, Vol. 27 No. 2, pp. 197-235.
- Rowlands, B. (2006) "The User as Social Actor: a focus on systems development methodology enactment", *Proceedings of the 21st ACM Symposium on Applied Computing, Dijon, France*, pp. 1540-1545.
- Rowlands, B (2008) "The Enactment of Methodology: an Institutional Account of Systems Developers as Social Actors", *Scandinavian Journal of Information Systems*, forthcoming.
- Scott, W. (2001) *Institutions and Organisations*, (2nd ed), Thousand Oaks, CA, Sage.
- Van Akkeren, J. and Rowlands, B. (2007) "An Epidemic of Pain in an Australian Radiology Practice", *European Journal of Information Systems*, Vol.16 No.6, pp. 696-711.