On Requirements Elicitation for Software Projects in ICT for Development

Kristina Pitula

A Thesis
In the Department
of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Doctor of Philosophy at
Concordia University
Montreal, Quebec, Canada

April 2010

© Kristina Pitula, 2010

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-71106-4
Our file Notre référence
ISBN: 978-0-494-71106-4

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Abstract

On Requirements Elicitation for Software Projects in ICT for Development

Kristina Pitula, Ph.D.
Concordia University, 2010

Currently, there is much interest in harnessing the potential of new and affordable Information and Communication Technologies (ICT) such as mobile phones, to assist in reducing disparities in socioeconomic conditions throughout the world. Such efforts have come to be known as *ICT for Development* or ICT4D. While this field of research holds much promise, few projects have managed to achieve long-term sustained success. Among the many reasons for this, from a software engineering perspective, in many cases it can be attributed to inadequacies in the gathering and defining of software requirements. Failures in realising sustainable systems stem from inadequate consideration of the high-level socioeconomic development goals, neglect of environmental constraints, and a lack of adequate input from end-users regarding their specific needs and sociocultural context. The situation is exacerbated by inadequate reporting on the social impact of such interventions, making it difficult to assess a project's success, let alone apply lessons learned to new projects.

In this thesis we propose enhancing conventional requirements elicitation with a complementary elicitation methodology specifically adapted to address these shortcomings. Our approach is based on a proposed novel technique of *Structured Digital Storytelling* to elicit input from end-users having limited literacy in the form of stories. The proposed methodology includes a systematic method for extracting and interpreting the informational content of the stories that applies a conceptual model derived from Communications Theory to identify constraints arising from the users' sociocultural context. The thesis introduces an *ICT4D quality model* identifying non-functional requirements related to the sociodynamics of a system's sustained use in a rural community. The needs, goals and constraints thus identified are integrated using a goal-based analysis to produce a more informed understanding of potential areas of technology intervention and to develop high-level functional and non-functional software requirements. The resulting goal model is also used in deriving a measurement framework for assessing a project's success based on its social impact. We illustrate our approach and validate its effectiveness with a field study.

Keywords: ICT4D, digital divide, requirements engineering, needs elicitation, requirements elicitation, culture, storytelling.

Acknowledgements

I would like to take this opportunity to thank the many people who have assisted me on this journey. My supervisor, Dr. Radhakrishnan, for his vision, guidance and infinite patience in keeping me on track and in scope. Dr. Dysart-Gale from the GSU, for our many enlightening interdisciplinary discussions on communication, technology and society. My committee members, Dr. Mudur and Dr. Grogono, for their insightful feedback and advice, especially during my research proposal. Dr. Reimer from the Department of Sociology and Anthropology, for his invaluable insights into community development and technology. NSERC (Natural Sciences and Engineering Council of Canada), FQRNT (le Fonds Québécois de la Recherche sur la Nature et les Technologies) and Concordia University for their financial support. I would also like to thank the many colleagues and friends who contributed to the cause: Daniel Sinnig, Richard Becker, Mary McDermott and Suzanne among others. To my family, and especially my partner, Frederik, for their unfailing support throughout these past few years. Finally, I extend thanks to the many "stakeholders" who participated in our projects and without whom this research would not have been possible.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

A growing world-wide effort is underway to provide disadvantaged people in developing countries with access to digital content and services using information and communication technologies (ICT). Such efforts are referred to by the term *ICT for Development* or ICT4D. Impelled by programmes such as the *Millennium Development Goals* (United Nations, 2009) and belief in the Internet's potential as a tool for social transformation, these projects seek ways of applying ICT to redress critical disparities in socioeconomic conditions throughout the world. While the dominant access model of the last decade was the rural telecentre, based on shared access by means of personal computers with landline connections to the Internet, there is currently a shift towards mobile devices. Recent developments in affordable mobile phones, low-cost wireless network technologies and Web 2.0 overcome many of the obstacles arising from the lack of adequate infrastructure, ICT capacity and relevant digital content that have plagued projects modelled on the telecentre, making their long-term sustainability and scalability questionable (Heeks, 2008). The rapid and wide-spread proliferation of mobile phones in developing countries, their low cost, portability and operability by non-literate people offer new opportunities for designing innovative, accessible, and cost-effective ICT4D solutions that, with suitable business models, can become financially self sustainable (Rashid & Elder, 2009). Regardless of the access model, harnessing the potential of these technologies requires developing appropriate software applications to deliver relevant information and services in a manner whereby the intended users can sustainably benefit. The nature of the targeted users and the types of problems they face make this a challenging software design problem that is far from being well understood let alone resolved.

The people targeted by ICT4D projects are the poor of the world. Sometimes referred to as the "bottom (or base) of the pyramid" (BOP), this group consists of the 3.7 billion people—or over half the world population[1]—that lives on less than US $8 per day and is largely excluded from the formal market system (World Economic Forum, 2009). Of this group, over 2.6 billion survive on $2 or less a day, with the "bottom billion" living in "extreme poverty" on less than $1. Although BOP income levels are found worldwide, they are mainly concentrated in Asia, Africa and Latin America, with 60% in India and China. However, poverty is not restricted to the BOP, as

---

[1] As of March 2010, the world population is estimated at 6.8 billion. (Wikipedia, retrieved March 31, 2010 http://en.wikipedia.org/wiki/World_population).

throughout the world there are many people whose income exceeds the BOP criteria but is still below the poverty line. Including these in the count swells the ranks of the poor considerably.

The social divide between the world's wealthy and poor is correlated to a "digital divide", referring to inequalities in the abilities of people to use ICT to access and benefit from digital information and services. While 20% of the world is now online, this is mainly in developed countries (United Nations, 2009). 80% of websites are in English, which is only understood by 10% of the world population[2] (Rice, 2003). Almost all technological innovations come from a geographic area representing 15% of the total world population. While some 50% of the world population is able to adapt these technologies for their own use, the remaining 35% is entirely disconnected. Thus, while ICTs are recognised as reshaping the flow of investment, goods and services in a global economy, it is only a small minority of the world population that actually benefits from this, leading to a polarisation whereby the rich become richer and the poor become poorer. The rapid expansion of new ICTs is likely to reinforce and exacerbate the existing socioeconomic inequalities if left to evolve on its own.

It is these inequalities that social development projects seek to redress, with programmes such as the *Millennium Development Goals* set by the UN member states to i.e. reduce poverty, provide universal primary education, promote gender equality and female empowerment, combat disease, and ensure environmental sustainability (United Nations, 2009). ICTs and mobile phones in particular, have a favoured position due to their perceived role as enabling tools, with applications such as m-banking and m-commerce on mobile phones, and disaster management upheld as examples. The vision put forward is to view the BOP as a business opportunity to develop in cooperation with the private sector, by creating "life-enhancing offerings"[3] and leveraging "hidden assets"[4] within the communities (United Nations, 2009; World Economic Forum, 2009). However, this vision requires qualification, as otherwise it easily slips into a techno-optimist view of ICT as a silver bullet. It is in this context that technologists and engineers, particularly software

---

2 Sources cited for these percentages were published in 1999 and 2001 (Rice, 2003).

3 "Life-enhancing offerings" refers to "offerings that improve the livelihoods of the BOP by pricing for their budgets, tailoring products to address local constraints, and developing environmentally sustainable approaches". (pp. 6. World Economic Forum, 2009).

4 "Hidden assets" refers to untapped resources within communities such as undocumented capital, community and personal resources, and underutilised assets (World Economic Forum, 2009).

engineers, must develop knowledge about the characteristics and constraints of the BOP so that they can develop successful and sustainable solutions from the perspective of the people that will use them.

With regards to actual ICT4D projects in the field, although numerous pilot projects have been attempted over the past decades, few have managed to bring long-term sustained benefits to the people that they target. Despite the often considerable hype surrounding their launch, many projects have failed to deliver evidence of any real social impact (Kleine & Unwin, 2009; Rashid & Elder, 2009; Heeks, 2008). A far too great emphasis on technical success with inadequate concern for the end-users' needs and the social development aspect of the projects are among the factors that have contributed to this lack of success (Heeks, 2008). In many projects, the multi-dimensional sets of goals and constraints that characterise ICT4D projects are inadequately addressed. In particular, many of the social, cultural and economic factors that affect the sustainable use of technology in a rural context are often overlooked. Frequently, existing technologies are introduced in a non-inclusive manner, without sufficient adaptation or reinvention with regards to the users' needs and sociocultural context (Frohlich et al. 2009). This deficiency can be overcome by involving the targeted beneficiaries (or end-users) more fully in elaborating project requirements, a strategy in line with the participatory approaches commended by development theory (Kleine & Unwin, 2009). However, because of their socioeconomic situation, end-users generally lack the literacy skills to be able to articulate their problems and needs in a manner amenable to conventional requirements elicitation approaches. Cross-cultural differences between the end-users and other stakeholders contribute to further misconceptions regarding their needs and how they might be effectively addressed.

1.1 Problem statement

ICT4D projects applied in the context of social development present numerous, complex challenges, many of which can only be addressed at a political level. We tackle the problem of designing technology to bring sustainable, measurable benefits to a rural community from a software engineering perspective. Many of the shortcomings with ICT4D projects described above can be reformulated as inadequacies in the gathering and defining of software requirements. We contend, based on well-established software engineering principles, that a clear statement of a project's high-level goals is essential in defining a software system's purpose. Equally essential is the early involvement of end-users in elaborating requirements, as this will lead to systems that satisfy their needs more fully, and satisfying these needs within the constraints imposed by the users' sociocultural context will result in more successful systems.

However, achieving effective communication between software analysts and end-users from disadvantaged rural backgrounds using conventional elicitation techniques is a major challenge. Additionally, there is no clear formulation of what the sociocultural factors affecting the sustainable use of technology in an ICT4D context might be and how they might impact a software system's design. Moreover, integrating information pertaining to users and their sociocultural environment into conventional requirements engineering (RE) processes is in itself an open research question (Cheng & Atlee, 2007; Nuseibeh & Easterbrook, 2000). These are the challenges we tackle in this thesis.

1. **Incorporating the social development goals in defining ICT4D project goals and measuring project outcomes.** While social development is *the* driving force behind ICT4D, with many ICT4D efforts the social development aspect is implicitly assumed. Few ICT4D efforts explicitly state the developmental goals being pursued or how they will be addressed. Often, there is an unstated assumption that because a project targets some disadvantaged group with some new technology that can produce a "positive result", it will have a positive social impact, without substantiating the claim or considering its sustainability, scalability or the form this impact will take let alone measuring the actual outcomes (Kleine & Unwin, 2009; Rashid & Elder, 2009; Heeks, 2008).

2. **Overcoming the barriers to involving end-users in elaborating project requirements.** Establishing effective communication with end-users is indispensable for involving them in requirements elaboration. A key characteristic distinguishing the end-users' sociocultural context from that of analysts is their literacy. Here by literacy we refer not to the basic ability to read and write, but rather to the associated analytical skills that allow people to analyse and express their problems and needs in abstract terms. Asking the 'right questions', understanding the 'questions right', and giving the 'right answers' are all learnt skills which, because of their socioeconomic situation people in rural communities may not have developed. Nor do analysts necessarily have the experience and skills to know what questions to ask, how to ask them and how to interpret the answers within a particular rural context. Differences in language and social position act as further barriers to effective communication between users and analysts. Consequently, rural people are likely to have problems articulating and communicating their information needs through conventional interviews or questionnaire media.

3. **Providing a more precise formulation of the social, cultural and economic factors affecting the sustainable use of technology in a rural context.** By its nature, the ICT4D

domain is multidisciplinary. Although there is a considerable body of research related to social development and the digital divide, it is largely framed in a social science perspective, providing little direct guidance applicable to designing ICT4D software systems. While there is a growing body of research specific to ICT4D, much of that literature is anecdotal and unsubstantiated, or focused on specific issues from particular disciplinary perspectives. To date, there is no comprehensive and succinct treatise on the multitude of interrelated factors affecting sustainable ICT4D projects and the software requirements therein. In particular, culture with respect to technology and specifically with regards to disadvantaged rural populations is inadequately treated.

4. **Integrating contextual information pertaining to ICT4D projects into conventional requirements engineering processes.** Reasoning about users and their context, and integrating such information into the requirements engineering process is recognised as difficult, even when dealing with conventional systems. In the case of ICT4D projects, the amount of contextual information to consider is augmented considerably (see points 1–3 above), with no clear way of linking that information to standard RE processes.

1.2 Research statement and objectives

Our overall area of research is ICT4D for rural communities with a focus on methodologies for specifying software requirements that lead to sustainable systems that make a real difference to their intended beneficiaries. The primary objective of this thesis is to present a requirements elicitation methodology that addresses the shortcomings of conventional elicitation approaches when determining requirements for ICT4D software projects. Such a methodology will assure that the critical factors with respect to a project's success and sustainability are systematically factored into its software requirements from the beginning. Currently no such methodology exists, and ICT4D software requirements (as well as the projects themselves), are elaborated in an ad-hoc manner when it comes to dealing with the particularities of the ICT4D domain. This is at the root of many of the observed failures. By defining a methodology that addresses the key challenges identified in the previous section, we hope to redress this situation, leading to more successful outcomes for such projects in the future.

Below we highlight the major contributions of this thesis. Each one engages a hitherto unaddressed aspect when it comes to determining software requirements for projects in the ICT4D domain.

1. **A participatory approach for elaborating ICT4D software requirements.** Drawing on a multidisciplinary body of literature relating to the digital divide, ICT4D, culture from a communication theory perspective and sociology, we have devised an approach for eliciting requirements directly from representative end-users that capitalises on their customary mode of expression. Our approach, specifically tailored to the characteristics of the targeted societies and the sociocultural dynamics related to literacy, enables rural people to express their "needs" in the form of "stories". Suitably interpreted, these stories provide software analysts with information on the users' activities, goals and concerns as well as the context in which they occur, imparting a more complete and accurate depiction of the problem to address and the requisite software requirements from the bottom up.

2. **Tools, models and techniques to support participatory ICT4D requirements elaboration.** In order to assist software engineers in determining requirements for sustainable ICT4D systems, along with the corresponding techniques, we have built a prototype tool for eliciting stories from rural populations and developed a set of models identifying the critical factors to consider with respect to the local rural context. These models synthesise the many factors involved in designing sustainable systems for a rural context, including those pertaining to the sociodynamics of sustainable ICT use and the cultural characteristics of the targeted users. Regarding the latter, we propose a model that provides insights into the deep aspects of culture and its impact on communication and affecting change in a society. The elements of these models are factored into the tool's design to provide a tool that is more than a simple recording mechanism.

3. **A methodology for converting "stories" into software requirements.** We present a methodology that augments the conventional RE process by collecting stories from end-users and then analyses and abstracts these stories into sets of needs, goals and constraints for a software system. We provide a systematic process for extracting and interpreting the stories' informational content, and for modelling the extracted information into an analytic representation suitable for software analysts. With our models, we identify additional needs, goals and constraints related to the users' sociocultural context that might otherwise go unobserved. Our methodology places user needs foremost in driving requirements at the same time that it identifies contextual information that might otherwise be overlooked, providing analysts with a more complete and accurate understanding of the problem and constraints from the users' perspective, and thus of the functional and non-functional requirements that a potential solution must satisfy. The output of this process serves as primary input to a

conventional RE process and for defining a set of measurements for assessing a project's social impact.

In order to test our elicitation technique and validate our methodology, we conducted an experiment in rural India. Using the prototype elicitation tool we developed, we conducted three field studies, eliciting information on two different topics in two different linguistic regions. To show that our methodology is effective in eliciting non-obvious and non-trivial information and to demonstrate how our methodology is applied, we present the results from the analysis of the stories collected in one of these studies.

1.2.1 Limitations of our research

Because of the nature of our research, empirical studies are necessary. In the case of ICT4D projects, this involves international collaboration with the villages in which experiments are conducted as well as the agencies involved in social development work, adding considerable logistic complexity to any endeavour. At the same time, the timeframe for our doctoral research imposes a limit on the number, scope and duration of the field studies that we can undertake, affecting what aspects of our proposed methodology we can test. To fit within the scope of a thesis, we have focused our experimental work on the elicitation aspect of our methodology and testing the needs elicitation tool. Thus, our experiment was not conducted in the context of an actual ICT4D project with external stakeholders and domain experts, and the implementation we developed is a proof-of-concept rather than a deployed system.

1.3 Thesis organisation

The remainder of this thesis is organised as follows:

In **Chapter 2**, we characterise the ICT4D domain and its challenges. We describe the social aspects of technology acceptance and use in rural communities, and consider the cultural differences between industrialised and developing nations before considering the relation between social development and ICT4D. We then reformulate the ICT4D challenges as a requirements engineering problem, and describe the shortcomings of applying conventional approaches to ICT4D projects. We conclude by defining the scope of the problem we address in this thesis.

In **Chapter 3**, we provide background information on requirements engineering, describing the core RE activities with an emphasis on elicitation. We review some conventional modelling techniques before presenting goal-based analysis as a way of relating requirements to concerns in the problem domain. We provide a brief overview of sociotechnical systems and some related modelling techniques. We then present the Goal-Question-Metric (GQM) approach as a means of defining a measurement framework for evaluating system success. We conclude with the challenges of requirements elicitation in an ICT4D context.

In **Chapter 4,** we present an overview of the theories and applications related to our work. We cover Participatory Rural Appraisal (PRA) from the field of social development, computer-based interviewing and storytelling from a sociological perspective. We briefly describe the modelling techniques applied by our methodology before presenting the dominant views on culture with respect to information technology and the theories underlying our own cultural model. Finally, given that the approach we propose is inherently qualitative, we briefly discuss qualitative and quantitative research methods.

In **Chapter 5**, we present the models and methods that constitute the building blocks of our requirements elicitation methodology. We first present a conceptual model of technology use in a rural society and the set of desirable properties or non-functional requirements that an ICT4D project should satisfy with respect to the local context. We then present our cultural model, describing the characteristics of an experiential society and the constraints to which it gives rise. We follow by describing our notion of *Structured Digital Storytelling* for eliciting needs from people with no or low literacy, and compare it to conventional elicitation techniques.

In **Chapter 6**, we present a detailed description of the SDS methodology for needs elicitation in an ICT4D context and explain how it fits within the core requirements engineering activities. We describe the process for identifying themes, collecting and processing the stories and modelling the extracted information, detailing the step-by-step process for extracting and abstracting the information and constructing the models. We then explain how the resulting artefacts are used in conventional requirements specification and validation, and for constructing a measurement framework for evaluating a project's success.

In **Chapter 7**, we present the E-Tool, a prototype tool we have developed to support the SDS elicitation technique. We explain the considerations that went into its design, its feature set and some proposed enhancements following our field tests as well as a projected future version and some potential applications.

In **Chapter 8**, we present the experiment whereby we validate the effectiveness of the SDS approach and demonstrate how it is applied. We first describe the context of our experiment before describing the experiment itself and its outcome. To show that our approach is effective and demonstrate how it is applied, we provide a detailed analysis of the stories collected in one of the studies.

Finally in **Chapter 9**, we summarise our work, highlighting our contributions and their positioning with respect to the social development discourse.

Chapter 2: ICT4D and Challenges

In this chapter we first provide a brief review of ICT4D efforts to date, characterising the projects and the challenges specific to them. We then consider the social aspects of the acceptance and use of technology in a rural context, emphasising the role that a "community of practice" plays in encouraging a community to become engaged and appropriate a technology for its own use. We next consider the cultural aspects of technology, emphasising the cultural differences that are likely to exist between the targeted end-users and the other stakeholders involved in a project, and among the stakeholders themselves. Following that, we consider the social development aspects of ICT4D projects and the considerations this introduces. We then reformulate the ICT4D challenges as a requirements engineering problem, focusing on the lack of precision in defining a system's purpose, inadequate consideration of local environmental (non-functional) constraints, the lack of end-user involvement in defining requirements and the lack of attention to a project's social impact. We conclude this chapter by defining the scope of the research problem we address in this thesis, mentioning some limitations of our research with regards to resolving the problems related to the digital divide.

2.1 ICT4D characteristics and challenges

The term ICT4D is used to describe a wide range of endeavours that have the common goal of promoting the socioeconomic development of disadvantaged communities through the direct or indirect use of ICT. These projects are driven by high-level social and economic development goals that most often are initiated from outside the targeted communities. The projects involve stakeholders from the public, private and non-profit sectors such as social workers, agronomists, and representatives of government, business, and international funding agencies as well as NGOs and local community initiatives, working in partnership in the most effective instances (Ramirez, 2001). The intended beneficiaries typically have limited schooling, low literacy and income levels, and only speak local languages. Many live in dire poverty with no obvious way of extricating themselves. The developing countries and regions where the projects take place are characterised by inadequate infrastructures, intermittent power and connectivity, underdeveloped economic markets and distribution and support networks as well as a lack of trained personnel. Often remoteness and extreme climatic conditions such as heat, cold, dust, or humidity introduce additional operating constraints. All these factors contribute to creating a novel context, far removed from that of conventional ICT applications.

There are three main thrusts to ICT4D initiatives:

1. developing infrastructure to provide power, connectivity and devices appropriate for the prevailing conditions;
2. building ICT capacity corresponding to the skills and competencies necessary to maintain and use the technology; and
3. providing digital content and services.

All three are essential for a project's success, and due to the prevailing conditions, many projects must address all three in unison. Converging on a combination that satisfies all the novel conditions and constraints that characterise these projects is complex. And even if a project is technically successful, there is no guarantee that its high-level social development goals will be attained. For this reason, despite the best intentions, many ICT4D projects have failed to bring long term sustainable benefits to the communities in which they are deployed. The following are among the reasons cited in the literature (Tongia & Subrahmanian, 2006; Unwin, 2009):

- Multiple stakeholders have vague and non-converging objectives, with little or no input from the ultimate beneficiaries
- Vague project objectives result in a lack of clear metrics for evaluating success, making claims of success largely dependent on which stakeholder defines it
- Deployment and sustained operation constraints are inadequately addressed, with the result that many projects do not survive beyond the prototype stage once external support is withdrawn
- Usability requirements and evaluations are inadequately reported making it difficult to assess how usable a project is by its target population, let alone apply lessons learnt to new projects
- Requirements pertaining to economic sustainability are not considered, limiting a project's potential adoption and dissemination

In the case of the stakeholders, their diverse backgrounds and areas of expertise often result in a set of disparate high-level goals. If these are incomplete or vaguely stated, they can easily be misinterpreted or overlooked. With regards to the environment, the technical, economic and cultural conditions which characterise the context-of-use introduce novel constraints that will compromise a project's success if not addressed. Devices must be appropriate for the prevailing operating conditions (climate, infrastructure, available support), and provide relevant services that are affordable and accessible to their intended users. Here economic and social factors come into play. For example, to keep costs low, shared facilities may be used. However this may introduce confidentiality and privacy concerns, and other social considerations such as access restrictions due to age, gender or social standing (e.g. women may not be able to visit sites frequented by men or people from lower castes may be denied access; Garai & Shadrach, 2006). And even if the services offered are relevant and accessible, other factors such as personal obligations, public opinion or local customs may prevent users from being able to fully benefit from them.

Determining what is relevant, accessible, and applicable requires the input of end-users, yet all too often they are not consulted when project goals are set. These are frequently established by external experts to comply with the agendas of national and international funding agencies, and elaborated in a top-down manner without full understanding of local conditions (Kleine & Unwin, 2009). Even if the villagers' input is solicited, their social status, limited literacy and lack of exposure to ICT act as barriers to their full participation using conventional elicitation approaches. When end-users are disconnected from a project's goals, they are likely to be unmotivated, distrustful or simply unable to make use of a technology.

Although technology is a core component of any ICT4D effort, experience has shown that the technical success of a project is not sufficient for a successful outcome. Here, by the term technology, we refer to the hardware devices, software applications, and physical infrastructure to access information and data services electronically. Also essential is the ability of people to use a technology in order to engage in meaningful and gainful social activities in a sustainable manner (Warschauer, 2003). According to the *Unified Theory of Acceptance and Use of Technology (UTAUT) model*, for a technology to be accepted by its intended users, it must be perceived as beneficial, easy to use, and socially endorsed, with an adequate infrastructure in place to support its use (Venkatesh et al., 2003). To meet these objectives, a technology must be relevant to the community's needs, build on existing knowledge and skills, and be affordable and sustainable. To be part of a sustainable cycle, the benefits from a technology's use must balance its costs. Finally, for a project to be economically sustainable it must produce a measurable outcome in a cost-effective manner, be scalable as the user population grows, and be maintainable after deployment (Koch & Caradonna, 2006).

These factors give rise to the following key challenges specific to ICT4D projects:

1. Success is to be measured by achieving sustained communal benefits that evolve over the long term as opposed to short term. Metrics to measure the resulting benefits are difficult

but necessary in order to show a compelling value proposition that justifies the funding needed to sustain a project beyond the prototype stage.

2. Deployment and sustained operation constraints cannot be resolved from a purely technological perspective, but are dynamically interrelated to a community's broader socioeconomic context. For the technologies to be sustainable in communities where widespread poverty is the norm, innovative business models are needed, and their requirements must be incorporated into the projects from the beginning.

3. There are major social, cultural, economic and political differences between "technologically developed" and "technologically underdeveloped" societies that impact their ability to make use of ICT effectively and sustainably in realising lasting changes; these differences reside in the social dynamics as well as structural characteristics of these societies.

2.2 Social aspects of technology

The differences between technologically developed and underdeveloped societies are at the heart of the digital divide. While there is a considerable body of research on the social aspects of technology acceptance and use in industrialised countries and particularly in organisational contexts, the social aspects with respect to non-technological societies have been less studied. In characterising the digital divide Warschauer (2003) has identified four barriers that prevent individuals or communities from being able to effectively use technology for accessing information and services. These barriers consist of access to: (a) the physical resources such as devices and infrastructure, (b) the digital information resources such as software and content, (c) the human resources corresponding to the skills required to find, extract and apply knowledge, and (d) the social resources referring to the broader social context in which technology use takes place. While ICT4D initiatives typically engage the first three barriers, namely ICT infrastructure, digital resources and capacity building, it is the fourth barrier that comprises the social dynamics that will affect a society's ability to effectively and sustainably benefit from a technology beyond the prototype stage.

While there are clear differences between rural and technologically developed societies, there is no simple definition of what these differences are. Markets, financial capital, manufactured goods, media-based culture and technology are generally situated in populated urban areas (Reimer, 2005). Such areas can be characterised in terms of their population densities, flows of people, flows of information and capital, and connectedness of inhabitants, with the location of people with respect to the flows based on a historical and cultural interpretation of the space. Participation in the urban dynamics entails mobility that creates a flow of "familiar strangers" providing individuals with anonymity and privacy (Williams & Dourish, 2006). Typically, the denser the population and more dynamic the flows, the more diversity, prosperity and potential opportunities present, with corresponding levels of economic and political power.

In contrast, rural communities generally have low population densities, limited flows and connections, and more static relations, leading to fewer economic opportunities, a slower rate of change and reduced anonymity and privacy. Because of distance and tradition, they are disconnected from political decisions. The low population densities and reduced flows of information and goods provide little incentive for infrastructural investment and limit exposure to new technologies and practices, including those associated with ICT usage. Here we introduce the concept of "community of practice" similar to the notion of apprenticeship, derived from the theory of *Situated Learning* (Lave & Wenger, 1991). This theory views learning as a situated activity where learners participate in a community of practice. Newcomers initially participate from the periphery, observing and assisting masters. As they gain mastery, they move from the periphery to a more participatory role, eventually becoming masters themselves. Without exposure to ICT practices, community members are likely to have difficulty imagining how some technology might be beneficial to them, let alone learning how to apply it themselves in order to realise benefits and create new opportunities. Consequently, when introducing a new ICT product, it is also necessary to address the presence of a related community of practice.

There are a number of elements involved in developing such a community of practice. According to Ramirez (2001), for a community to become engaged and develop the skills and local support necessary to appropriate a technology for its own use, the following elements are required:

- the emergence of a team of local champions
- a community based organisation that responds to the community's vision
- a trusted learning 'space and place' where community members can explore a technology's benefits and limitations
- policy and funding support as a community's capacities and needs evolve
- close working relationships between champions and policy makers that allow them to learn and adapt to one another

These elements cannot be imported but must be developed locally so that the technology is grounded in the community's experience and integrated in its daily activities. The local champions provide the impetus for developing a community vision while the community based organisation facilitates its realisation. The learning space and place is the venue for developing a community of practice with respect to applying that vision and evolving it, provided adequate funding and policy support is available. As the team's experience, skills and social capital grows, the sophistication of the projects undertaken can evolve accordingly. It thus becomes incumbent on ICT4D projects to not only introduce a technology, but to also foster a community of practice whereby people can see how they might benefit from that technology, learn to apply it, and evolve the associated practices to address emerging needs within their local context.

2.3 Cultural aspects of technology

The underdeveloped countries and regions in which ICT4D projects typically take place make culture a key factor. Culture is a collective phenomenon that shapes the attitudes and behaviours shared by all social groups, touching upon all aspects of daily life. The concept of culture is formalised and interpreted in a variety of ways, covered later in chapter 4. Franklin (1990) describes culture as the set of socially accepted practices and values shared by a group of people, with practices "*the way things are done*". Practices are the observable manifestations of a culture expressed through symbols, artefacts, and procedures varying from forms of discourse, dress and art to societal structures, methods, laws and rituals. Values, in contrast, are largely unobservable, consisting of the set of knowledge, beliefs, norms of behaviour and ways of thinking that underlie the practices and give them meaning (Kersten et al., 2002).

For a technology to be accepted and used, along with the socioeconomic changes it is intended to bring, it must fit within a community's value system and be in harmony with local cultural practices. At the same time, there are likely to be significant cultural differences between the external stakeholders involved in a project and the targeted end-users, and among the stakeholders themselves, with values and practices drawn from different nations, organisations and disciplines. These differences will extend to the implicit values and practices embedded within conventional software processes and applications, the majority of which have been designed by, and for an Anglo-American organisational context (Boehm, 2006). These differences encompass the values and practices associated with the integration and use of technology in daily life. When assumptions from the industrialised world are applied to developing country contexts, discrepancies are likely to occur.

Heeks (2002) refers to such discrepancies as "design-actuality gaps", corresponding to a disconnect between the analyst's vision of how a software system should operate and the end-users' local reality. Such gaps arise when analysts make invalid assumptions regarding the users' context, relating to things such as people's activities, their goals and underlying values, or the types of information available, where it resides, and how it flows, is transformed and used. There may be significant differences in the way workplaces and institutions operate, in the generally available technological infrastructure and in the meaning ascribed to technology itself. Many "best practices" advocated by conventional software approaches and tools embed implicit assumptions regarding "proper" software operation, based on conceptual frameworks and normative models that do not take into account local contextual particularities (Avgerou, 2001). Most prevalent is the assumption of "rationality" that pervades the software discipline (Avgerou, 2001; Heeks, 2002; Kersten et al., 2002). Heeks identifies a design-actuality model with seven dimensions to characterise the discrepancies that may exist between "rational design" assumptions from the industrialised world and the "local actuality" of developing countries, presented in Table 1. Note that the distinction between industrialised and developing countries is not a dichotomy, but rather two ends of a continuum.

Table 1. Differences between rational design and local actuality of developing countries

Dimension	Rational design assumptions	Local actuality
Information	Emphasis on standardised, formal, quantitative information distributed through formal channels	Contingent, informal, qualitative information circulating through informal channels
Technology	Simple enabling mechanism	Complex value-laden entity perceived as a status symbol or alternatively, tool of oppression
Processes	Stable, clear-cut, formalised processes with rational decisions based on logical criteria	Informal, flexible, complex processes with constraints and decisions based on criteria that are not strictly "rational"
Objectives and values	Formal organisational objectives	Multiple, informal and personal objectives with values drawn from the sociocultural context
Staffing and skills	Viewed as rational entities in sufficient number and with sufficient competency levels	Viewed as people within a social network
Management systems and structures	Emphasis on formal, objective processes and structures	Informal, subjective processes and structures
Other resources: time and money	Applied to attain organisational objectives	Applied towards personal objectives

(Adapted from Heeks, 2002)

In the case of the beneficiaries targeted by ICT4D projects, because of their socioeconomic situation the gaps are likely to be wide. The community's geographic constraints and social characteristics and dynamics will affect what information is accessible to its members, how it circulates and what is trusted. The technological infrastructure that is readily available and with which people are familiar is likely to differ radically. Additionally, the symbolic meaning people ascribe to ICT, often associated with progress and social status, will attract interest at the same time that it may induce anxiety, unrealistic expectations or misappropriation by those in a position of authority. Community livelihoods, based on traditional activities, will involve practices with deeply entrenched beliefs and values that are potentially in conflict with the changes that the project is intended to introduce. When it comes to institutions, the beneficiaries' low social status and marginal position put them at a disadvantage. In many developing countries, corruption, discrimination and unequal power relations are common and culturally reinforced, with those in authority having a vested interest in maintaining the status quo (Beardon, 2006). Non-profit and charitable organisations, while interested in making a social impact, will also apply their own cultural values with regards to what form of "social improvement" they endorse (Knack & Rahman, 2007). All these factors are potential sources of conflict with respect to the cultural values and practices espoused by community members, preventing them from being able to effectively benefit from the intended improvements.

2.4 Social development aspects of ICT4D

Understanding the nature of the intended "social improvements" and what form they take makes social development a critical dimension of ICT4D projects. Whether undertaken by the public, private or non-profit sector, social development programmes are a cornerstone of ICT4D work as they provide the funding, resources, contacts and field personnel necessary to evolve a proof-of-concept into a viable ICT4D service. These programmes may apply different forms of development, depending on their mandate and motivations. One form of development is equated with economic growth, with social improvements a consequence of improved economic conditions (Kleine & Unwin, 2009; Trainer, 2002). Another form focuses on capacity building, and developing the skills and competencies to participate in the economy (Prakash & De', 2007). Yet another form views development as a process of empowerment whereby the marginalised gain power and self-determination on matters affecting their lives (Beardon, 2006). Here, development promotes assisting people in developing the skills, motivation, know-how and confidence to take charge of their lives, with social improvement a consequence of this process.

Within the field of development studies, a number of approaches are applied (Kleine & Unwin, 2009). Development through **modernisation** was the prevalent theory of the last century, founded on the naïve belief that prosperity could be achieved by modernising economies and societies through the introduction of "better" technologies that would bring increased economic growth and productivity along with the requisite cultural changes. While successful to a certain degree in certain contexts (e.g. India and China) and leading to some beneficial programmes, modernisation is not a panacea, as testified by the persisting and growing inequalities as well as the many well-intentioned but rarely used or abandoned ICT4D projects. Development through modernisation applies a traditional top-down, urban-centred information flow model, with knowledgeable experts propagating messages deemed valuable and appropriate for the poor by those in authority. At a theoretical level, modernisation is now considered patronising, and has largely been replaced by **participatory approaches** that promote the involvement of local community members in developing local solutions, with a variety of forms in use (Mohan, 2006). However, despite this shift in theory, the majority of ICT4D efforts are still based on a top-down development model (Kleine & Unwin, 2009).

ICTs can fulfill a number of roles in the social development context. In their primary role, they are a simple enabling mechanism that supports the delivery of digital information and services that are of value to a community at a given time. At the same time, they are also a tool of modernisation and a symbol of progress. As an enabling mechanism, they can deliver information and services that are empowering or they can reinforce existing power inequalities (Beardon, 2006). As a modernisation tool, they can improve existing practices, introduce new and complementary ones, or alternatively, they can introduce new practices that render the existing ones obsolete (Babe, 2000). In all cases, they are a symbol of progress that can have positive or negative connotations. What role they play is largely a function of the purpose to which they are put, with this purpose dictated by the many stakeholders both directly and indirectly implicated in a project. Consequently, an ICT4D project cannot be extracted from the political development discourse in which it takes place, and software analysts must be cognisant of this discourse if they are to concretise a project's purpose.

2.5 Reformulating ICT4D challenges as a requirements engineering problem

We consider the problem of developing successful ICT4D systems from a Requirements Engineering (RE) perspective. We contend that by addressing the shortcomings with current approaches for collecting requirements in an ICT4D context, we can make progress in addressing the key challenges identified above, namely measuring success based on realising sustained social benefits, resolving deployment and operational constraints with respect to the broader socioeconomic context and considering the sociocultural dynamics with respect to ICT acceptance and use. We focus our attention on the elicitation and analysis of user needs where we identify the following issues:

1. Understanding and incorporating the diverse goals of the different stakeholders to converge on a single, agreed upon set of achievable information system and service goals for the software project
2. Identifying all the environmental constraints that will impact project goals
3. Getting input from the targeted end-users with respect to project goals
4. Identifying high-level social development goals as the means of measuring project success

When multiple stakeholders from different areas of expertise are involved in a project, their goals will reflect diverse concerns and motivations. Stakeholders from different socioeconomic backgrounds will bring different perspectives and express themselves in different ways. These needs and concerns, along with their underlying assumptions, must be elaborated in detail so that they can be mapped into operational project goals and constraints, and any conflicts identified and resolved. This is essential in order to converge on a single set of operational goals that drive the software requirements and according to which the project's success will be measured.

The novel context in which ICT4D projects take place introduces new environmental constraints that may render established software solutions ineffective. Understanding what these constraints are and their impact on a potential solution is critical. It is also very difficult, as many of these constraints arise from sociocultural conditions and practices specific to these rural communities and foreign to the ICT4D analysts whose knowledge of the rural context and the concepts of poverty, non-literacy, and powerlessness are largely theoretical. Such sociocultural factors fall outside the scope of conventional requirements gathering, and thus are frequently overlooked.

With regards to getting input from end-users, along with overcoming the barriers of language, social class and literacy, it is also necessary to consider the cross-cultural differences between the targeted society and that of the ICT4D practitioners. As people lack exposure to ICT, they are unaware of the potential benefits and limitations of such technologies, and how these might be made relevant to their needs. Consequently, end-users are unlikely to be able to speak of their needs in terms of technological interventions. Similarly, practitioners from outside the community, although aware of the potential benefits of technology, are unlikely to be familiar with that society's precise needs, making it difficult to probe them in depth. Somehow it is necessary to reconcile these two views so that practitioners can develop an informed view of the problem to solve, and the potential solution addresses the needs of the majority of users within the constraints imposed by their sociocultural context.

When it comes to measuring success, evaluation must go beyond establishing a project's technical success to measure how well it attains its social development goals. However, because of the time-frames required by a target society to achieve results, this is difficult. External stakeholders are generally interested in showing positive results as rapidly as possible to justify themselves with respect to their funders. Consequently, project evaluation often takes place with a focus on technical achievement shortly after a project's deployment, when people are still enthused by its novelty and before any social impact or steady-state usage can be attained. If any subsequent long term evaluation is conducted, as with any business, projects that prove to be unsuccessful simply fade from public attention. Yet measuring a project's ability to deliver sustainable long term benefits is critical both for extracting lessons from the experience and for obtaining support to evolve a project beyond the prototype stage to widespread deployment. Therefore, a project's social development goals must be placed front and foremost in terms of driving the development effort and evaluating the subsequent deployment in both the short and long term. Stating these goals as project acceptance criteria and planning for their evaluation upfront focuses attention on them and ensures that they are precisely formulated.

2.6 Scope of the problem addressed by this thesis

The ICT4D arena presents numerous, complex challenges. The field of socioeconomic development of which ICT4D is an off-shoot, is in itself a wicked problem that is far from being resolved. An ICT application on its own cannot redress inequalities in the power relations and distribution of wealth that are at the heart of the social divide characterising the ICT4D field. Consequently, ICT4D applications must play a supporting role within a broader development initiative that generally involves researchers and practitioners from multiple sectors and disciplines, extending across the social, economic and political sciences into agriculture, medicine and business as well as the funding agencies (Dias & Brewer, 2009). ICT4D efforts are often characterised as multidisciplinary, with each discipline contributing its own particular expertise to address the problem at hand. However, in the case of establishing software requirements for ICT4D projects, the effort is intrinsically interdisciplinary as analysts play a central role in negotiating the different perspectives, terminology and expectations of the various stakeholders in order to converge on a single, coherent set of goals describing the system's purpose. To this, requirements analysts add their own expertise with regards to the development, acceptance and use of software technology. Thus, analysts require a judicious understanding of the many factors involved in an ICT4D project in order to design a software system that can effectively contribute to improving the situation of those it is intended to benefit.

Our area of research is requirements engineering for rural ICT4D software projects with the goal of designing technology that brings sustainable, measurable benefits to rural communities. The specific focus of this thesis is on collecting and analysing requirements with respect to end-users and their sociocultural context-of-use, and linking the information collected to conventional requirements engineering processes. While the following areas are of critical importance to the success of an ICT4D project, they are not directly addressed in this thesis:

- **The larger socioeconomic development context in which an ICT4D project takes place.** We do not consider the politics of development in terms of government policy, institutional support, or the ideologies underlying project funding and the business of international development. Instead, we assume that projects are undertaken in good faith, with sound intentions and adequate support with regards to a primary goal of improving the end-users' situation.

- **The elicitation of requirements from "other stakeholders" involved in a project.** By "other stakeholders" we refer to the diverse group of representatives from the different sectors and disciplines. We do not investigate how this diversity might affect requirements elicitation. Instead we assume that the requirements of these other stakeholders can be elicited using conventional techniques.

- **The business aspects of ICT4D projects in terms of appropriate business models for economic sustainability**. For a proof of concept project to achieve continuing success over the long term, economic sustainability is essential. If external funding is limited or not available, then a project needs a compelling value proposition, which together with an effective business model, can lead to wide-scale adoption and diffusion over time. Because of conditions in developing countries (i.e. the absence of infrastructure, an underdeveloped market, lack of distribution and support networks, etc.), this value proposition must cover operation and maintenance costs through self-generated revenues. This requires innovative business models such as tiered pricing geared towards the poor, micro-financing, partnering, franchising, etc. (Koch & Caradonna, 2003), and the requirements of these models must be incorporated into the overall system design. Although we consider affordability and sustainability as desirable project properties from the end-users' perspective, we do not address the business aspects with regards to generating revenue and cost-effective operation.

- **The philosophical debate regarding the neutrality of technology**. In this thesis, by emphasising the sociocultural aspects of technology use and acceptance, we implicitly adopt a holistic as opposed to reductionist position with regards to the relation between technology and society (Kersten et al., 2002). In other words, we posit that a society cannot be detached from its culture, which is what ascribes meaning to its existential manifestations, and that technology, viewed as both the tools and practices whereby a society "does things" is simply another cultural manifestation embodying a society's value system and knowledge—in direct opposition to the theory of technological determinism, which views technology as independent of the society from which it arises, and thus neutral (Franklin, 1990). We support our position by referring to Heeks' (2002) design-actuality model describing the disconnects between industrialised and developing countries, and we refer to Avgerou's (2001) characterisation of software related "best practices" as based on the assumption of "rationality", but do not address the philosophical debate per se in this thesis.

- **The debate regarding the validity of quantitative versus qualitative research and scientific knowledge**. As this debate has been dispelled by the scientific community at the forefront of elaborating and applying these research methodologies, it is not addressed here. A brief discussion is presented in chapter 4, with a short description of each methodology and their respective merits and disadvantages.

Chapter 3: Requirements Engineering and Project Success Metrics

Requirements Engineering (RE) is a crucial step in the development of any software system. It is the process whereby the intended purpose of a system is discovered and documented so that it can be analysed, communicated and eventually culminate in a software implementation that meets that purpose. How well that purpose is met is the primary measure of a system's success. Thus RE is essential in determining what a system will do and how this will be measured. The process is inherently iterative, and consists of three core activities: elicitation of needs, requirements specification and requirements validation. The process starts with some ill-defined 'ideas' of what the system should do. These are elicited, analysed, and systematically transformed into a technical requirements specification that defines the software system to be built completely and unequivocally. Modelling plays a central role in representing, analysing, elaborating and communicating requirements among the stakeholders and developers, while the ability to link the stated needs to the technical requirements helps ensure that all the needs are met without superfluous features. The RE discipline offers a wide range of established methods and techniques for accomplishing the various activities, appropriate for different problem domains and development styles.

Below, we first provide some definitions pertaining to requirements and describe how they are conventionally represented before providing background information on the requirements process and the elicitation, specification and validation activities, with particular emphasis on elicitation. We then provide a brief overview of the role of modelling with respect to requirements, describing some common modelling approaches and presenting a goal-oriented approach in detail. Given the direct relation between a project's requirements and measuring success, we follow with the Goal-Question-Metric approach for evaluating the quality of a software product. We conclude with a discussion of some of the challenges of RE with respect to ICT4D projects.

3.1 Requirements

Requirements engineering is defined as "the branch of software engineering concerned with the real-world goals for, functions of, and constraints on software systems [and] with the relationship of these factors to precise specifications of software behaviour (Zave)" (pp. 37, Nuseibeh & Easterbrook, 2000). These factors (goals, functions and constraints) drive the identification of requirements to satisfy them, with "requirement" defined as a condition or capability that is needed by a user to solve a problem or achieve an objective, or that must be met or possessed by a system (IEEE standard 1233-1998). Effectively, RE is concerned with providing a precise definition of the problem, based on which a software solution can be defined. Thus RE can be considered the link between the real-world problem and the proposed software solution, with requirements largely representing the problem space and how the environment is to be affected while the other software engineering artefacts represent the solution space and how the solution is expected to behave (Cheng & Atlee, 2007). The purpose of RE is to discover and trace a mapping from the problem space to the solution space.

As the generic term '*requirements*' encompasses a range of related concepts, we clarify the meaning we assign to the various terms in common use. Requirements are conventionally referred to by the terms user needs, system features, constraints, properties, and software requirements per se, with each pertaining to a specific category of information regarding the problem space. *User needs* refers to the complex set of exigencies on the intended system arising from the users' explicit and tacit goals, desires, expectations, activities, immediate context-of-use and broader organisational and social setting, with *user* (also referred to as stakeholder) encompassing both the end-users and other stakeholders implicated in the system. *System feature* refers to a service the system will provide to fulfill one or more user needs. *Constraint* refers to a condition or restriction affecting one or more features with which the system must comply, while *property* is a quality that the system must possess. *Software requirement* refers to a precise capability or condition that the system must fulfill in realising a specific feature or constraint, with one or more requirements related to a given feature. All can be expressed at different levels of abstraction. Whereas user needs essentially reflect the problem from the users' perspective, the system features, constraints and software requirements all provide a product perspective in that they describe the product's external behaviour with respect to the problem.

In defining the problem space, there is a natural, orderly progression from user needs, to system features and constraints, to precise software requirements (Leffingwell & Widrig, 2003). Each constitutes a step in the mapping from the real-world problem towards a software solution. There is a comparable progression in the precision with which they are conventionally expressed, as exemplified by the templates associated with the *Rational Unified Process* (RUP), a widely accepted industry standard for software engineering (Rational Software Corporation). In RUP, user needs are presented in a succinct problem statement of predefined format[5], complemented by

---

[5] RUP suggests the following predefined format for the problem statement, with blanks to be filled by the system analyst: "The problem of ... affects ..., the impact of which is ... A successful solution would ..." (Rational Software Corporation).

a cursory description of the users, their profiles, environment, responsibilities with respect to the system, and an itemised list of needs. While this information is directly related to the system features in that it provides the rationale for their inclusion, it is rarely complete with respect to the full set of data gathered on users, and even less so regarding the real-world situation, often leaving developers with a skewed view of the users' real needs (Saiedan & Dale, 2000). System features provide a high-level description of what service the system will provide in general and concise terms, and system constraints are similarly described. Once the feature set and constraints have been established, the specific software requirements associated with each feature are specified in sufficient detail to provide a contractual description of the feature's external behaviour. Here we introduce the notion of "well-formed requirement". This is defined as "a system functionality (a capability) that can be validated, and that must be met or possessed by a system to solve a customer problem or to achieve a customer objective, and is qualified by measurable conditions and bounded by constraints" (pp. 11, IEEE standard 1233-1998). This set of requirements is often augmented by a supplementary use-case specification that describes the sequence of actions or interactions between a system and user for accomplishing some outcome of value.

From a software engineering perspective, requirements are classified into two broad categories: (1) a *Functional Requirement* (FR) identifies a specific need the system must satisfy or action it must be able to perform. Each FR is specified in terms of the outputs produced from given inputs, and when combined together, show the relationship between a system's inputs and outputs; (2) a *Non-Functional Requirement* (NFR) specifies a property the system must possess or constraint it must meet. With regards to the relationship between FRs and NFRs, FRs state <u>what</u> the system will do, whereas NFRs constrain <u>how</u> that what is to be achieved. In other words, NFRs define properties and/or constraints that introduce restrictions on the FRs' behaviour. Each FR describes a specific function of the system and can be considered self-contained. In contrast, NFRs define global constraints on the system which may affect multiple FRs, and each FR is likely to be constrained by multiple NFRs. Although NFRs are expressed as general properties, they effectively introduce constraints on specific FRs, and the operationalisation of general NFRs with respect to high-level FRs leads to more specific FRs in the solution space. When considering NFRs, there are a number of requirements models, such as *FURPS+* and the *ISO 9126* software quality standard, that define categories of requirements considered standard properties for software projects (i.e. functionality, reliability, usability, efficiency, maintainability, portability, etc.).

Documenting requirements is essential to their effective communication and management. Being able to track requirements as they emerge and evolve over time is of critical importance. For this it is necessary to identify and document the relationships between requirements in their different representations so that they are traceable across the entire lifecycle, from the original 'raw' needs to the various software engineering artefacts to which they lead. As a result requirements must be represented in a form that is easy to read, navigate, query and change in both a forwards and backwards direction so that the consequences of changes can be analysed and propagated at any stage (Nuseibeh & Easterbrook, 2000).There is a wide variety of document formats and templates to choose from, and a number of software tools expressly designed for this purpose. User needs and system features and constraints are generally recorded in a *Vision* document, while the detailed requirements are recorded in a *Software Requirements Specification* (SRS). A separate *System Requirements Specification* is used to document requirements pertaining to the overall system, including hardware, software, people and operating procedures.

3.2 The requirements engineering process

The RE process fits within a larger software engineering process of which a large variety are available. Consequently, while the purpose remains the same, the terminology and partitioning of the RE activities may differ somewhat depending on the particular conventions used. In this section we focus on the RE process itself, and its three core activities, namely elicitation, specification, and validation. The purpose of elicitation is to arrive at an understanding of the real-world problem and define the users' needs, based on which a set of system features and constraints is identified. During specification, these features and constraints are elaborated into well-formed software requirements, while validation ensures that these requirements are consistent, complete and accurate both internally and with respect to the stated user needs. These three activities are intertwined, as the elicitation of needs leads to the specification of requirements, in the course of which ambiguities and omissions arise. These are clarified through further elicitations, leading to additional requirements. The same occurs during validation, and these activities are repeated until a complete and validated set of well-formed requirements is attained. While the major RE effort occurs early in the software development cycle, requirements change in the course of development and evolve after deployment, making requirements an element that spans the life-time of a software system. The core activities along with their key inputs and outputs are depicted in Figure 1 and described in the following sections.

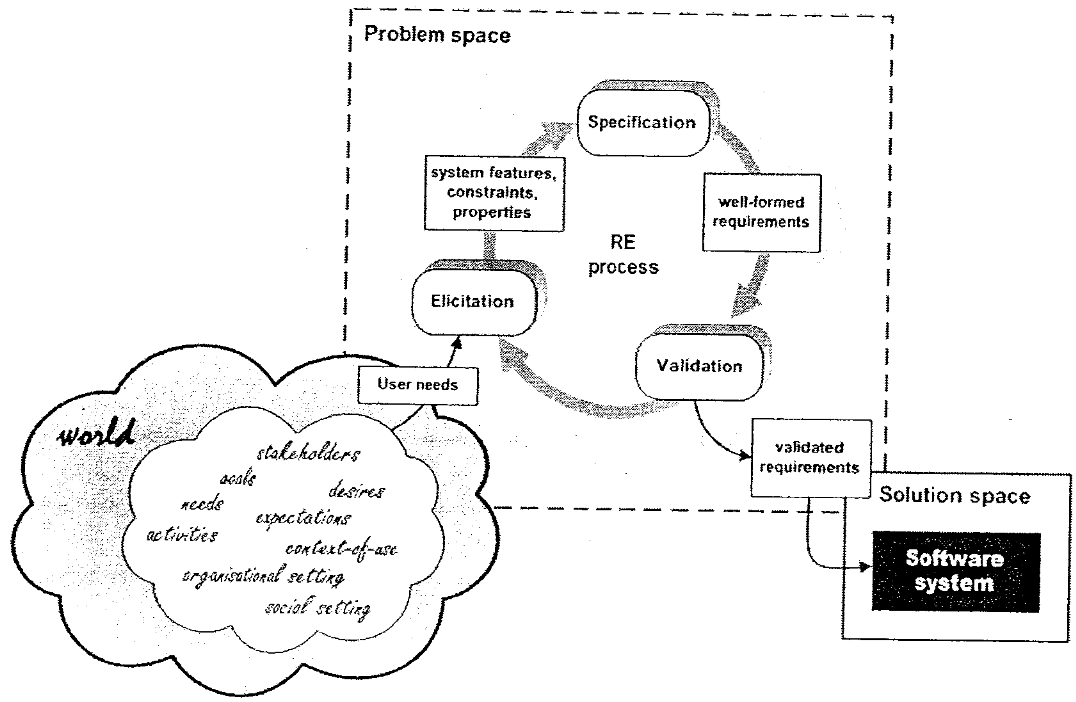

Figure 1. Core requirements engineering activities with key inputs and outputs

3.2.1 Requirements elicitation

Requirements elicitation consists of the activities necessary to understand what problem the proposed system will address, where the system boundaries lie, who the stakeholders are and what they need in order to derive a set of features, constraints and properties describing the system to build. This activity generally starts with some preparatory groundwork to establish a high-level vision for the project and assess its feasibility and associated risks. While this preliminary step is not always considered part of elicitation (and sometimes is not explicitly documented), it constitutes the initial 'idea' that drives all the subsequent activities, including the choice of elicitation strategy. Once a high-level direction has been set, the elicitation process itself is highly iterative, and proceeds in both a top-down and bottom-up manner. The problem is sketched out, key stakeholders are identified and their needs elicited, solution boundaries are drawn and constraints identified. These factors are interdependent, as how the problem is defined and where the system boundaries are drawn will influence which stakeholders are implicated and their needs, affecting the definition of system features, constraints and properties. The discovery of impediments in elaborating these factors can lead to their revision, including the project's high-level vision and feasibility.

Given an initial, imprecise idea of the system's purpose, elicitation starts with defining the high-level system objectives, identifying the stakeholders, and analysing their needs, motivations, and activities with respect to the objectives, in order to delimit and define the problem space. While sometimes the problem definition appears clear-cut, more often it isn't—what appears to be the problem may simply be a symptom of a deeper one that is the 'real' problem that needs to be addressed (Leffingwell & Widrig, 2003). Elicitation is a process of discovery that relies on the collection, analysis and interpretation of data obtained from stakeholder interviews, on-site observations and the study of pertinent background materials in order to ensure that the 'right problem' is being addressed and that it is thoroughly understood. This is not straightforward, as stakeholders often have vague, conflicting goals, needs and expectations that must be drawn out. Goals and needs must be prioritised, and conflicts negotiated and resolved before a single problem definition that satisfies all the stakeholders can be attained.

When identifying stakeholders it is important to consider the full cast of people both directly and indirectly implicated in an intended system. Among others, this cast includes champions, funders, end-users, indirect users, maintenance personnel, domain experts and the software engineers responsible for development as well as everyone who will sign-off on the delivered system (Saiedian & Dale, 2000; Leffingwell & Widrig, 2003). Champions are those who take a proactive interest in realising the project vision. These are the people who provide access to resources and eliminate obstacles, clearing a path through the problem domain, and thus can have a significant influence on the form a solution takes. Not all projects have champions. Funders provide project financing, and will have their own motivation for pursuing a project as well as practical interests in terms of budgets, timelines, etc. End-users are those who will interact with the system either directly or through an intermediary, and will be the stakeholders most familiar with the work practices supported by the system as well as those directly affected by reliability, availability and usability concerns. Other users may have a direct interest in the system outputs but won't actually use it themselves. Domain experts bring in-depth knowledge regarding the problem domain and system environment, while software engineers bring technical expertise regarding software development. The people who will maintain the system will also have their own specific needs. Finally, the people who sign-off on the project will have their own delivery criteria. Each group will bring its own motivations, skill set, background and viewpoint to bear upon the project. Depending on the circumstances, one person may fill several roles, or alternatively one role may be filled by one or more groups of people. Not all projects involve the full cast of stakeholders described above, while others may involve additional players such as marketing, sales, business development, etc. Establishing who the stakeholders are is crucial, as omitting a key stakeholder may comprise the success of a project.

Given the stakeholders' diversity, elicitation is understandably difficult (Saiedan & Dale, 2000; Coughlan & Macredie, 2002). Analysts want to obtain knowledge about the problem domain that is relevant to the design. At the same time stakeholders may not fully understand the problem themselves, they may not know what is relevant, they may not know what they want or have difficulty articulating it. Many people, while readily able to talk about their activities, are not necessarily aware of everything actually involved in performing some activity, nor capable of taking the next step to reflect on what they do, let alone the following step of identifying what they might need or want. Such tacit knowledge is of high value to analysts but also the hardest to elicit. Stakeholders may have unstated expectations, they may be resistant to the changes the new system embodies, and thus uncooperative or evasive, and some may simply have invalid knowledge. Stakeholders and analysts may lack the appropriate knowledge or shared understanding to communicate effectively. They may use different terminology or jargon, or make invalid assumptions. Analysts might not ask the 'right' questions or omit relevant topics, while stakeholders may not give the 'right' or complete answers. Finally, their respective attitudes may create an atmosphere of discomfort or distrust. For all these reasons it is important to devise an elicitation strategy that takes into account the stakeholders' characteristics and the nature of the knowledge sought as well as the analyst's facilitation skills.

Devising an elicitation strategy consists of selecting appropriate participants and suitable techniques to obtain relevant information most effectively while ensuring that pertinent information is not overlooked. Once the key stakeholders have been identified, there is a plethora of elicitation techniques from which to choose (Nuseibeh & Easterbrook, 2000). These include traditional techniques such as surveys and structured or unstructured interviews, and group techniques such as brainstorming, focus groups, and requirements workshops. Prototyping (both low and high fidelity) can be used to obtain early feedback from users and to trigger discussion, and is often used for exploratory purposes. In certain situations, cognitive techniques such as think-aloud protocols, card sorting and role playing can provide insight, while other situations may call for techniques borrowed from knowledge engineering. Contextual techniques study user needs with respect to a particular context and environment to help ensure that the future system is appropriate with respect to the users' characteristics in that environment. Commonly used techniques for this include techniques borrowed from ethnography and user and task analysis. A given strategy will likely combine several techniques appropriate for eliciting different types of knowledge from different stakeholders, and the results from one round of elicitation can readily lead to another. Analysts must be flexible in adapting their strategy to probe unforeseen issues and pursue promising leads.

Among the RE activities, elicitation is recognised as being the most critical and difficult. It is critical because its outcome determines what the system will actually do and thus the system's ultimate success or failure. It is difficult because it is a dynamic problem solving activity involving discovery, interpretation, and negotiation to converge on a definition of the problem that satisfies all the stakeholders. The output of elicitation is a suitably documented set of user needs and related system features, constraints and properties that will serve as input to requirements specification.

3.2.2 Requirements specification

During specification, the features, constraints and properties derived in the course of elicitation are elaborated into "well-formed" software requirements. Following the guidelines for developing system requirements specifications described in (IEEE Standard 1233-1998), high-level system features are decomposed into their constituent requirements, with each requirement expressed in terms of the inputs received, the function performed and the outputs produced as well as its operational conditions and constraints. These correspond to the system's functional requirements (FR). Constraints and properties are refined into non-functional requirements (NFR) expressed in measurable terms. While properties are generally expressed as qualities of the system as a whole, they can be related to the specific functional requirements on which they have a bearing, as explained below. Additional properties from standard requirements models, such as reliability, usability, efficiency, maintainability, portability, security, etc. are introduced and similarly expanded. The requirements are analysed to ensure that they are "well-formed" and contain no open questions or possibilities for misinterpretation. They are examined for anomalies such as unexpected interactions between requirements, potential obstacles or unstated assumptions, and the discovery of any is likely to lead to another round of elicitation for clarification purposes.

Requirements are the primary means for communicating the stakeholders' needs to the technical community responsible for developing the software system. Consequently, they must be expressed with technical precision, and formalisms such as UML (Larman, 2005) are commonly used for this. The output of the specification activity is a *Software Requirements Specification* (SRS) that unambiguously depicts the set of functional and non-functional requirements for the envisioned software product.

3.2.3 Requirements validation

Requirements validation consists of ensuring that the SRS accurately reflects what the stakeholders need and want, and is consistent and complete. One facet of accuracy involves ensuring that the specified requirements address the stated user needs (quality of conformance) while another considers whether the specification fulfills the users' actual needs and expectations (quality of design). Conformance is established by ensuring that each stated user need is addressed by at least one requirements statement, and that superfluous system features unrelated to any need are excluded. Establishing a specification's consistency consists of ensuring it contains no unexpected interactions or conflicts between requirements. Formally expressed requirements can be checked using formal verification techniques such as static analysis or model-checking (France & Rumpe, 2007). However, these cannot be used to evaluate the quality of design, and formal specification languages are arduous, limiting their use. With semi-formal or informal specifications, validation is accomplished using informal techniques such as walkthroughs, reviews, and checklists (Leffingwell & Widrig, 2003). In all cases, quality of design can only be assured by having stakeholders review and approve the specification. Consequently, the requirements need to be expressed in a form that stakeholders can readily understand. The validated requirements approved by the stakeholders constitute the contractual agreement between the stakeholders and software project team regarding the software system to build and its expected performance, against which the delivered system will be tested prior to its official acceptance.

3.3 Modelling requirements

Modelling plays a key role throughout the software engineering process. Models are abstract graphical representations that synthesise data describing complex situations to simplify their description and highlight their critical characteristics. Modelling techniques can be applied at different levels and stages of the software lifecycle, to model the problem, the software system, or the larger sociotechnical system comprised of both social and technical elements. There is a vast array of informal, semi-formal and formal modelling techniques (and supporting tools) for creating descriptive, predictive and even executable models, as well as model driven techniques whereby abstract design models are systematically transformed into software implementations. Different techniques provide different views of the system being modelled, and thus support different ways of reasoning about it. Here we focus on modelling with respect to elicitation, to assist in the elicitation process and to represent the ensuing requirements as understood from the elicitations. In this regard, models provide an abstract representation of the entities, relationships, behaviours and constraints of the problem, and serve as a tool for elaborating, analysing and communicating requirements among stakeholders and developers.

Within the software engineering discipline, the Unified Modelling Language (UML) is the industry standard (Leffingwell & Widrig, 2003). Originally developed to model software, its use has been extended to model the structure and dynamics of the problem as well as the broader context in which the problem is situated, with the latter commonly referred to as "business modelling". UML models provide the basis for communicating a shared understanding of the problem, based on which application requirements and downstream software artefacts are derived (Larman, 2005). Commonly used models for representing the problem include:

- *Domain models* – describe the important concepts within the problem domain, and their relationships and attributes. They are often represented using ER notation or UML class diagrams.
- *Information or data flow models* – describe how information or data flows through the system, and how it is transformed by the various processes therein. This view focuses on the functions performed in the domain, and the data that these functions consume and produce. These can be represented using UML dataflow diagrams or activity diagrams.
- *Control flow models* – describes the dynamic behaviour of the system in terms of the different states it can assume, and how it transitions from one state to another. These are commonly represented using state chart or UML state transition diagrams.
- *Task models* – provide an alternative perspective that focuses on the users' activities, describing the tasks they perform in terms of the sequence of steps required to complete some task and the dependencies among the steps. These are often used to represent the interaction between a user and the system. UML use-case models provide one such representation by means of scenarios (or use-case instances) describing specific sequences of actions and interactions, while a number of other tools offer graphic visualisation.

3.3.1 Goal-oriented requirements engineering

While for certain problems the above models are adequate, their primary focus is on data and processes. They do not explicitly capture the rationale behind the system, making it difficult to relate requirements to high-level concerns in the problem domain. A goal-based analysis is an established technique for explicitly doing so (Van Lamsweerde, 2001). Goals are objectives for the system to achieve, describing some intended capability or property of the system, while constraints are properties and conditions that must be satisfied in achieving those goals. Goals drive the identification of requirements to support them. Starting with high-level goals that identify the purpose of the system, these are successively decomposed until a set of functional requirements by which these goals can be met is attained. This is both a top-down and bottom-up process in the course of which goals are refined and abstracted, interdependencies and constraints are identified, alternatives are considered and conflicts resolved. By asking "how" and "why" with respect to already identified goals and requirements, more goals or alternative sets of requirements for achieving the same goals can be identified. A goal-refinement tree provides a traceable structure for representing the relationships among the different goals, subgoals and constraints, and for linking high-level goals at the top of the structure to operational requirements at the leaves.

Chung et al.'s (2007) *Non-Functional Requirements* or *NFR Framework* is one such structure for modelling system properties (or NFRs) as goals. Because NFRs are by nature subjective and relative, they are represented as *softgoals* to *satisfice* (i.e. satisfy sufficiently), rather than regular goals with a binary true or false value. *Softgoals* may interact with each other producing side-effects, and they may contribute positively or negatively, and fully or partially in achieving other softgoals. The NFR framework uses labelled arcs to indicate AND/OR relationships, conflicts and positive or negative support between the goals and with respect to the constraints. One of its purposes is to operationalise general NFRs with respect to high-level FRs in order to produce more specific FRs in the problem space. The framework includes catalogues of NFR templates providing knowledge about specific NFRs for reuse. The template for a given NFR sets out the concepts and terminology for expressing the associated requirements, implicit interdependencies with other NFRs as well as established development techniques. The operationalisation of an NFR is achieved through incremental construction, elaboration, analysis and revision of a *Softgoal Interdependency Graph* (SIG). A high-level SIG pertaining to security is depicted in Figure 2.

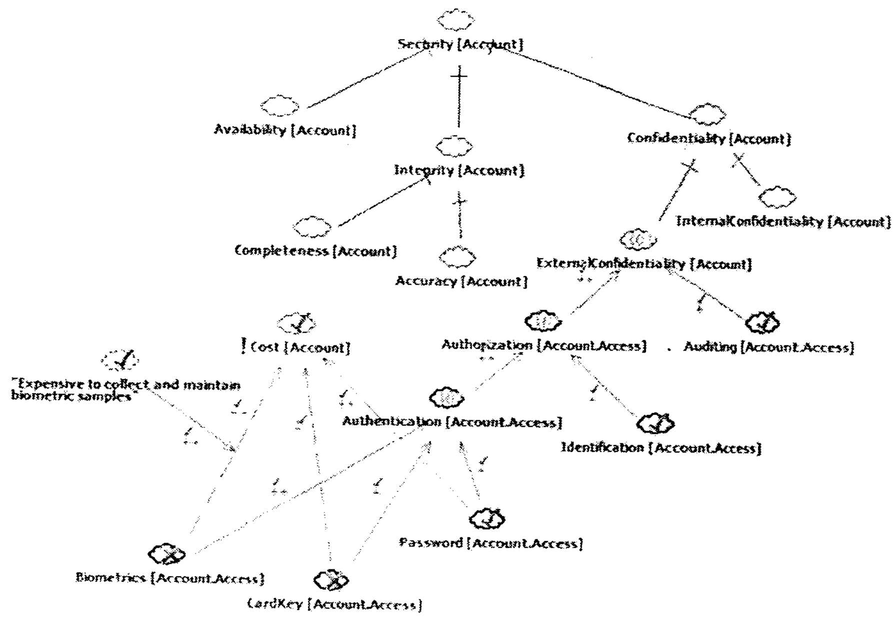

Figure 2. Softgoal Interdependency Graph pertaining to security (Supakkul, 2010).

3.3.2 Sociotechnical systems

Many problems require more comprehensive consideration of the environment and behaviour of people with respect to the software system (Cheng & Atlee, 2007; Nuseibeh & Easterbrook, 2000). A sociotechnical systems approach offers an alternative view that incorporates such considerations. Rather than focusing on technology alone, the problem is viewed in terms of the interaction between people and technology. Such an approach is particularly useful for representing complex systems in which organisational, human and software actors are interdependent and rely on each other to realise their respective objectives. The design of work processes for such systems requires joint optimisation of the interaction among the elements in order to make the overall system more effective and acceptable (Bryl & al., 2009). Sociotechnical systems are studied from both a social sciences and engineering perspective, with the latter focusing on understanding the nature of the interactions in order to determine requirements for the software components. While there is growing interest in the area of sociotechnical system engineering, methods and models for representing and analysing their requirements remains an open research question (Cheng & Atlee, 2007; Jones & Maiden, 2005; Suttcliffe, 2000).

A sociotechnical system generally contains three types of elements: (1) technical elements such as devices and software actors; (2) non-technical elements such as human actors; and (3) organisational elements pertaining to the social infrastructure in place that imposes such things as rules, standards and practices (Ottens et al. 2005). The functionality of the system is largely determined by the interdependencies among its elements, with dependencies that may be physical, functional, intentional or normative. For example, elements may enter in physical contact with each other, they may perform functions for one another, they may have specific intentions or goals with respect to each other, or they may impose norms by which other elements must abide. By identifying the set of dependencies among the actors which, if respected, fulfill all stakeholder goals, the objectives of the overall system can be attained. This set of dependencies corresponds to the requirements of the sociotechnical system (Bryl et al., 2009).

The modelling of sociotechnical systems is an open research area. Because the notion of "agent" captures the intentional aspects of human and software actors, agent-oriented approaches are often applied. The i*/TROPOS framework (Chung et al., 2000) is one such approach that builds on the concepts of the NFR framework to model organisations as networks of social actors with softgoals to satisfice, goals to achieve, tasks to perform, and resources to furnish. Using the i* framework, the relationships among actors are represented using a *strategic dependency model* that shows the dependencies among actors with respect to each others' softgoals, goals, tasks and resources. A related *strategic rationale model* similar to the *softgoal interdependency graph* shows the reasoning behind the relationships in terms of positive and negative contributions among the softgoals, goals, tasks and resources. The i* approach is supported by a number of specialised automated tools. Other methodologies exist, such as RESCUE (Jones & Maiden, 2005), that incorporates the modelling of human activity, goals (using the i* approach), and use case models with systematic walkthroughs of the scenarios, or Gaia and ROADMAP (Bryl et al., 2009), both of which emphasise the actors' goals and roles. However, integrating behavioural and contextual information about people and their environment remains an open issue when identifying requirements.

3.4 A Goal-Question-Metric approach for measuring project success

Given that the primary measure of a system's success is determined by how well it meets its intended purpose, there is an obvious link between a system's stated goals and its success metrics. Among the methods available for defining a measurement framework to assess system quality, the *Goal-Question-Metric* paradigm (GQM) is one of the most widely used and is considered a de facto standard (Scholtz & Potts Steves, 2004; Berander et al., 2006). GQM is a top-down approach which starts by explicitly defining measurement goals. These goals are then refined into a set of quantifiable questions, which in turn are refined into a set of objective or subjective metrics for which data is collected. The outcome of this process is the specification of a measurement model that targets a particular set of issues, linked to a set of key questions that characterise each issue in a meaningful and measurable way, ensuring that the measurement data collected is relevant and effective with respect to the stated goals.

The basic GQM model is a hierarchical structure consisting of three levels (Basili et al., 1994). In (Potts Steves & Scholtz, 2005), the authors present a 5-level evaluation framework that builds on the basic GQM model by introducing levels of abstraction. These additional levels serve to partition the measurement space and combine measures into an overall evaluation as well as distinguish between conceptual and implementation-specific elements, allowing for the definition of an evaluation template that can be used to collect comparable metrics for different systems or for the same system in different studies. The 5 levels consist of: (1) system goals, (2) evaluation objectives, (3) conceptual metrics, (4) conceptual measures and (5) implementation specific measures. Environmental factors that influence an experiment are documented separately, allowing for reuse of the same goals and objectives in different experimental settings. This 5-level evaluation framework is depicted in Figure 3 and described below.

The **system goals** at the top (level 1) consists of the set of high level organisational or performance goals specifying the desired benefit or functionality that the system is expected to deliver. This goal can relate to an organisational goal, such as improving a process or product, or to the system's technical performance. Each system goal is refined into a set of evaluation objectives at the next level.

The **evaluation objectives** (level 2) related to a given system goal, represent the different concerns that will be evaluated with respect to that goal, with each objective corresponding to one concern. Ideally objectives are chosen to cover all aspects of the system goal with minimum overlap between them.

Next are the **conceptual metrics** (level 3). Here a distinction is made between "metrics" and "measures", with a metric defined as "the interpretation of one or more contributing elements that can be other metrics or measures". Thus a metric can provide an overall evaluation for a set of measures, or for a set of other

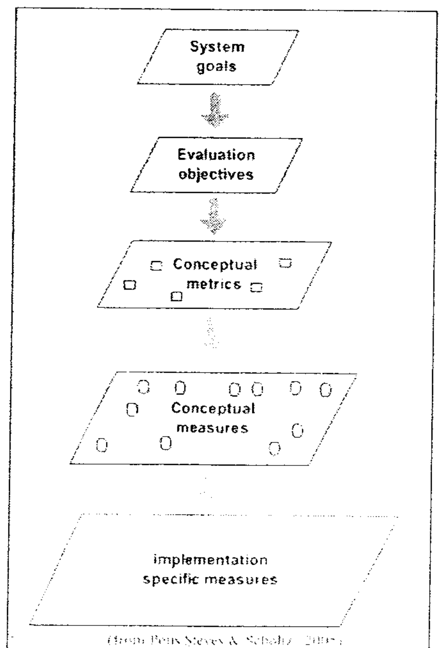

Figure 3. 5-level GQM framework

metrics, with its value based on a computation or an expert assessment.

A **conceptual measure** (level 4) is defined as a "performance indicator" that can be singly or collectively observed, and that can be computed, calculated, or collected automatically. A further distinction is made between conceptual and **implementation specific measures** (level 5), allowing for the same conceptual measure to be collected despite differences in the implementation.

Using a GQM approach, a measurement framework can be defined to assess a project's social development goals based on its high-level goal model. By including this framework in a project's evaluation plan one can assure that the social development aspects are not inadvertently overlooked when the deployment is assessed. While it does not guarantee that an assessment will be conducted after long-term usage, its existence makes it possible to do so even if the original stakeholders are no longer involved. Additionally, by placing emphasis on the social development assessment and defining how it will be measured early in a project's life-cycle, the project team will be encouraged to consider the project's purpose as well as its technical success throughout development. Moreover, by defining measures that are independent of the implementation, comparisons can be made between alternative implementations for attaining the same social development goals.

3.5 Requirements elicitation challenges

RE is a decisive and indispensible stage in the engineering of any software system, including those intended for ICT4D. It is also recognised as being one of the most difficult engineering tasks (Pressman, 2005). In the industrialised world, the mismanagement of requirements is among the leading reasons that software projects are problematic or fail (Taylor, 2000), with a lack of user input, incomplete and changing requirements, and unclear objectives among the chief contributing factors (Boehm, 2006). Incomplete and incorrect requirements inevitably propagate into the later stages of software development, leading to implementations that do not meet their users' needs. Such failures can be avoided by involving end-users and other stakeholders early in the elicitation process. This is recognised by the engineering community, and the involvement of stakeholders is a well-established principle of software engineering. However, effective communication between stakeholders and analysts is acknowledged as difficult (Coughlan & Macredie, 2002; Saiedian & Dale, 2000). Additionally, modelling the social and physical environment of a software system and integrating such contextual models with conventional RE specification and analysis techniques is an open research problem (Cheng & Atlee, 2007; Nuseibeh & Easterbrook, 2000).

In the case of ICT4D projects, understanding the users and their context is of critical importance. At the same time, the characteristics of the targeted user populations make it difficult to apply standard elicitation techniques. Furthermore, linking information gathered using contextual enquiry techniques to conventional RE processes is not evident. To date, although there is a growing body of research related to software development and deployment across national boundaries (Boehm, 2006), and global requirements management is receiving increasing attention (Cheng & Atlee, 2007), little work has been done in the area of cross-cultural requirements gathering, particularly with stakeholders from disadvantaged socioeconomic backgrounds. While sociotechnical systems approaches offer a more comprehensive view of people and their environment in relation to technology, the techniques are geared towards analysing situations in which there is a high degree of interaction between the social and technical elements. In contrast, with ICT4D projects, while understanding the social context is critical, the complexity resides in discovering the specification of the problem and the actual software system is relatively straightforward once the problem is understood (Dias & Brewer, 2009).

It is these challenges that we address with our proposed methodology, specifically designed for discovering and documenting the problem specification in an ICT4D context. We provide models to assist in identifying the critical environmental constraints, with particular attention to the cultural differences that may exist between rural and 'technology savvy' stakeholders, and the qualities an ICT system should possess. We offer an elicitation technique that is especially suited for gathering information on ill-defined problems from participants with low literacy levels. In addition, we provide a process for modelling and linking the elicited information to conventional RE specification and analysis techniques. Moreover, the goal model produced by this process serves as the basis for defining a measurement framework for assessing a project's social development goals for inclusion in the project evaluation plan.

Chapter 4: Related Fields and Foundations

The concepts we apply in our research are drawn from multiple disciplines. This chapter provides a brief overview of the theories and applications related to our work. From the social development field, we look at *Participatory Rural Appraisal* as a means of engaging participants in locally sustainable action. In the field of software development, we mention the use of computers to conduct structured interviews with end-users. We discuss storytelling from a sociological perspective, and describe its application in a range of domains—to develop a collaborative bottom-up analysis that serves as catalysis to action, for information sharing, and to collect contextual information. We then consider modelling techniques for representing and reasoning about problems, and briefly cover Domain Models, Influence Diagrams and Causal Loop Diagrams to represent and reason about a social system's static structure, causal relationships, and dynamic behaviour and feedback mechanisms. We consider culture with respect to software, where we present some of the prevalent views before expanding on the theories underlying our model. Finally, as our proposed SDS approach is inherently qualitative, we briefly discuss the nature of qualitative and quantitative research and their underlying values as well as their respective strengths and weaknesses.

4.1 Participatory Rural Appraisal (PRA)

In social and economic development circles the need to involve intended beneficiaries in development projects has long been recognised. The term *Participatory Rural Appraisal* (PRA) based on the work of Chambers (1994) is used to describe a variety of approaches that have evolved to facilitate the engagement of local people in development efforts. Here emphasis is placed on involving local people in analysing their situation with the goal of empowering them to plan and act on their own behalf in producing sustainable local action and institutions. The approaches described by Chambers rely on interviews, focus groups and community meetings to mobilise people and collect information using techniques such as participatory mapping and modeling, transect walks, matrix scoring, well-being grouping and ranking, institutional diagramming, seasonal calendars, trend and change analysis, and analytical diagramming—all undertaken by local people, often for eventual submission to the funding agencies. PRA approaches are primarily geared towards identifying and introducing social and economic interventions that local people can undertake on their own. Consequently, while they may play an important complementary role, they are not directly applicable to determining software requirements for suitable ICT tools.

4.2 Computer-based interviews

Hands et al. (2004) present a computer-based interviewing tool developed to facilitate the elicitation of information from end-users when gathering requirements and conducting user tests. In this work, the interview tool is primarily a means for automating conventional structured interviews. Analysts prepare a set of pre-defined and focused questions, which the tool then presents in a prescribed order. In this context the computer acts as a "non-threatening, non-judgemental interviewer with limitless patience", overcoming issues such as user inhibitions or a lack of personnel with appropriate interviewing skills. Although the authors initially envisaged collecting computer interviews to prepare for actual face-to-face meetings, they also tested the tool as an alternative to face-to-face interviews. Their experimental results showed that the tool was convenient and effective in eliciting useful information from users, and that users preferred the tool over paper questionnaires or phone interviews.

While this work is similar in spirit to our approach, we differ in how information is elicited. Rather than using structured interviews with focused questions that ICT4D users are likely to have difficulty answering, we apply open-ended storytelling on a set of predetermined themes.

4.3 Storytelling

Storytelling as a technique is applied in a wide range of domains. Stories constitute an art form, a form of entertainment and a fundamental mode of communication in use for millennia. In the social sciences, oral histories are used to provide alternative views on historical events based on first-hand experience, to capture cultural information, and to explore social issues. In business, storytelling is viewed as an integral part of organisational knowledge management (Snowden 1999), while in software engineering stories in the form of scenarios are used in the design process to communicate among stakeholders and developers. More recently, storytelling has been put forward as a means of eliciting requirements in domains such as healthcare, where access to end-users and the actual context-of-use is restricted (Gausepohl, 2008). Below we briefly describe this work.

From a sociological perspective, storytelling is an interactive, communicative activity that takes place between the narrator and an audience in the immediate present through the spoken word enriched by intonation and gesture. Storytelling varies from highly spontaneous and interpersonal accounts, such as those narrated around a dinner table, in the course of which everyday events are jointly constructed, deconstructed and reconstructed, to highly stylised presentations where revered stories are repeated almost word-for-word. In all cases, stories are a means of imbuing order and meaning out of the daily experiences of life. As such they convey important insights into how participants attribute meaning to their daily experience and identify themselves within their social world; and at the same time they reveal the culturally embedded normative influences under which they live (Orbuch, 1997).

In sociology, storytelling has proven itself as an effective means for expressing community information, issues and frustrations as the basis for developing a collaborative analysis from the bottom-up. In a recent study by Kerr (2003) it was applied to identify the problems of homelessness as perceived by the homeless themselves. The resulting analysis revealed a number of significant issues that do not emerge from conventional top-down analyses where input is solicited from such people as social service providers, public officials and academic experts. There was a comparable divergence in the nature of potential solutions and associated issues as viewed from the top-down versus bottom-up. Moreover, by moving from stories centred on life histories to stories concerning what could be done about the present situation, and broadcasting these stories to a wider audience, Kerr's research process of "telling and listening" to stories served as a catalyst to the homeless to become active in changing their situation.

In our ICT4D research context, we propose digital storytelling as a means for sharing information among semi-literate people in rural villages. A study by Frohlich et al. (2009) has shown the viability of storytelling as a means of communication in rural India. The study showed that villagers were enthusiastic about creating and listening to stories. The study also revealed a certain tension between those interested in creating and disseminating serious "development" content, and others more interested in creating personal and cultural content for entertainment purposes.

More recently, storytelling has been proposed as a means for eliciting requirements in domains where access to end-users and the actual context-of-use is restricted. In (Gausepohl, 2008), the author describes a field study in the healthcare domain comparing the requirements elicited using focus groups and interviews to those elicited using focus groups and stories. The objective of this study was to determine if there exists any difference in the number, breadth and depth of themes addressed and the amount of time required by participants. The study concluded that there was no significant difference with respect to the number and breadth of themes addressed. Moreover, storytelling was more effective than interviews in eliciting more diverse context-of-use and social information and required less time.

4.4 Modelling techniques

Modelling is both a tool and technique for representing and reasoning about problems. Models provide an abstract representation of the entities, relationships and behaviours that characterise some phenomenon. The modelling activity itself is a creative process that serves to develop an understanding of the phenomenon under study, to identify and represent the relevant concepts, to predict how the system behaves under different conditions and/or interventions, and to communicate these ideas to others. For our purposes, we are interested in graphical models that assist in conceptualising complex, real-world situations and are comprehensible to a general audience. Domain Models, Influence Diagrams and Causal Loop Diagrams are all established techniques that provide different ways of looking at the static and dynamic structure of a system as well as its feedback mechanisms, and in their semi-formal form, provide useful descriptions readily understood by people untrained in their use. Below we briefly describe these different modelling techniques and what each contributes.

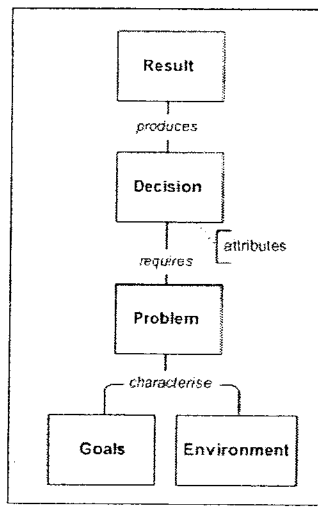

Figure 4. Domain model

Domain models are widely used to represent the important concepts within a domain, how the concepts are related and their attributes. They are useful in identifying and organising the various concepts a particular domain encompasses, and for establishing a common terminology for describing it. However, while the overall view they provide is valuable for understanding the static structure of a domain, it does not express the causal relationships or dynamics of the system. Domain models are commonly represented using ER (Entity-Relationship) diagrams or the more specialised UML (Unified Modelling Language) class diagrams favoured by software developers. The basic ER notation consists of rectangles to represent the entities, connected by labelled arcs representing the relations, as illustrated in Figure 4 This diagram can be interpreted from the bottom up as follows: given an environment and set of goals, these characterise the problem to solve, requiring a decision regarding the action to take that produces the result.

Influence Diagrams (Howard & Matheson, 2005), originating in the field of decision analysis, provide concise graphical representations for reasoning about the flow of information in decision situations. They provide an intuitive way to identify and display the essential components of a problem, namely the objectives, the uncertainties, the decisions, their outcomes, and how they influence each other. The basic notation consists of: hexagons to represent the variables to optimise; rectangles for decisions, ovals to represent uncertain variables, and double ovals for functionally determined variables. Arrows entering a decision node indicate the information available for making that decision while exiting arrows indicate the decision's influence. Arrows between uncertain nodes indicate relevance, i.e. the information from one node informs the other. An example is provided in Figure 5. Here, the environment and set of goals "inform" (i.e. define) the problem, which constitutes the information available to the decision, which in turn influences the result to optimise. While influence diagrams are highly useful for analysing the structure of a decision problem in terms of interdependencies among its components, by definition they cannot contain cycles, and therefore are unsuitable for expressing feedback mechanisms within a system.

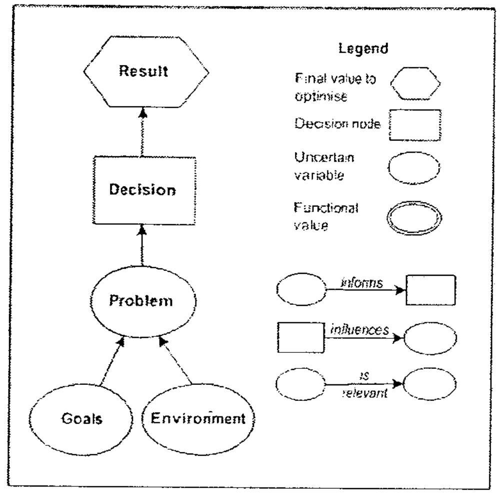

Figure 5. Influence diagram

Causal Loop Diagrams (CLD) (Sterman, 2001) from the field of Systems Dynamics are used to model complex, non-linear systems with feedback loops. Among their many uses, they are applied in the field of Systems Thinking (Forrester, 1994) to model social problems and their underlying causes in order to reason about the consequences of potential interventions to effectuate social change (Hirsch et al., 2007). With a systems perspective, problems are viewed in terms of feedback processes that give rise to problematic behaviour patterns. Here the system structure is described in terms of its constituent elements, interrelated by circular rather than linear cause-effect chains. These exert a positive or negative influence that respectively reinforces or undermines some desired situation. Such positive or negative feedback loops comprise higher conceptual units for describing a system's dynamic behaviour. The CLD notation consists of *elements* linked by arrows (called *causal links*) labelled with "+" or "–" to indicate that they produce a change in the same or opposite direction, with "||"indicating a delay before the effect is perceived, as illustrated in Figure 6

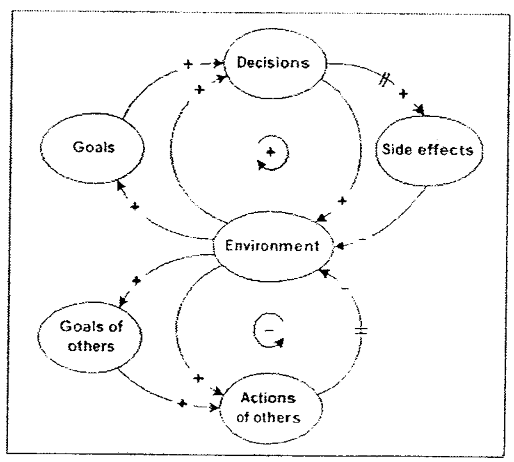

Figure 6. Causal loop diagram

(adapted from Sterman, 2001). This diagram can be interpreted as follows: the environment motivates an individual to define goals that lead to decisions regarding a course of action that will change the environment to some desired state. However, these decisions may also trigger unanticipated side-effects that will, after some delay, exert a negative influence on the changed environment. Seeing the changed environment, other agents with their own goals will react to restore the situation, to which the individual will react by redefining his goals, and so forth in an ongoing cycle.

4.5 Culture and software

The concept of culture is recognised and formalised by people for use in a multi-facetted manner. It is difficult to define precisely, and has a wide range of interpretations from different perspectives, each with its own terminology, purposes and traditions. The anthropologist Clifford Geertz defines it as "*an historically transmitted pattern of meaning embodied in symbols, a system of inherited conceptions expressed in symbolic forms by means of which [individuals] communicate, perpetuate, and develop their knowledge about and attitudes towards life*" (Gullivan & Srite, 2005). In the Informatics and Management literature, the most commonly cited definition is that of Hofstede (2005), who summarises it as the "*collective programming of the mind which distinguishes the members of one group or category of people from another*". According to Franklin (1990), culture is the set of practices and values shared by a society, with practices the observable ways things are done, ranging from forms of discourse, dress and art to societal structures, methods, laws and rituals. Values, in contrast, are largely unobservable, consisting of the set of knowledge, beliefs, norms of behaviour and ways of thinking that underlie the practices and give them meaning (Kersten et al., 2002). Below, we briefly describe some of the dominant views on culture and technology, before presenting the theories underlying our model.

While culture has been studied from a sociological perspective for a long time, it is only recently that globalisation has brought it to the forefront in the field of Information Technology. Currently there is much interest in the impact of culture on software as it is recognised that differences in national cultures and values can have a significant impact on software product and process adoption rates (Boehm, 2006). Given that the majority of computer devices, software applications and technology practices have been developed for use in an Anglo-American culture, there is growing awareness of the need for cross-cultural localisation to make them suitable for other cultural contexts, with a corresponding body of research which examines cultural factors with respect to *Software Localisation*. Its focus is on taking existing software products and adapting them to make them suitable for other countries. The related field of *Software Internalisation* is concerned with designing software applications that can be readily localised without requiring engineering changes (Esselink, 2003). A limitation of this research is its focus on the "external manifestations" of culture (such as language, currency, symbols, presentation formats, conventions, standards, laws, infrastructure etc.), with inadequate consideration given to the deeper aspects of culture (Kersten et al., 2002).

This neglect of "deep culture" is rooted in the underlying assumption prevalent within the software engineering community that cultural factors only affect the user interface and that core functionality and logic are culturally neutral. This assumption leads to the oversimplifying (and reductionist) view that "*all cultural aspects are encapsulated in the external layer of software*" (emphasis original), and can be localised by simply changing the user interface (Kersten et al., 2002). The fallacy of applying this view indiscriminately is most evident in the areas of ERP (Enterprise Resource Planning), GSS (Group Support Systems) and Collaborative Software, where corporate mergers and expansion have led to the deployment of such systems across organisational and international boundaries. The need to integrate the business processes and practices from different cultures has drawn attention to the implicit assumptions embodied in the technologies, and put these fields at the forefront of research on culture and IT adoption and use (Gullivan & Srite, 2005).

While the literature offers a variety of models for studying culture, these mainly consider culture from a national, ethnic, or organisational perspective. Hofstede's (2005) Cultural Dimensions (masculinity, power distance, individualism, long term orientation, and uncertainty avoidance) is among the most frequently cited, although his model has recognised limitations. Foremost among these recognised limitations are: the use of the nation-state as unit of analysis; its disregard of cultural differences that occur within or transcend national boundaries; its disregard of multicultural influences; and its view that culture is static over time, contrary to the now dominant view in anthropology that considers culture as emergent and dynamic (Myers & Tan, 2003). Additionally, there is no clear mapping between the cultural dimensions and operational requirements for a software system. Although other models exist, a common theme they share is examining culture in corporate or business environments with sophisticated, urban populations very different to the rural populations targeted by ICT4D projects.

ICT4D projects typically take place in underdeveloped countries and regions, making culture a key factor. As the field of *Human-Computer-Interaction* (HCI) deals with the human issues with respect to computer technology, by convention culture is generally examined from an HCI perspective, with emphasis on the input to and output from the computational elements, and how these interactions fit into the broader context of use. The HCI community has a rich repertoire of proven methodologies and techniques for doing this. To understand the users and tasks they perform, ethnographic studies are generally used. Such studies consist of going onsite to talk with users and observing their activities and behaviours related to the proposed system's functionality. This includes observing what supporting artefacts they use as well as the environment in which the activities take place. These studies can be quite intensive in terms of both the extent of the interviews conducted and the amount of observations made. However, when these activities take place within the framework of an actual project, practitioners rarely have the time or resources to conduct comprehensive, in-depth analyses. Traditional ethnographic studies give place to "rapid ethnography" (Millen, 2000), with the risk that these lapse into "scenic fieldwork" (Dourish, 2006), summarised as "I went there and this is what I saw". Such studies are likely to reveal only the surface manifestations of culture, as the larger social fabric in which they are embedded goes largely unobserved and the deeper cultural meanings cannot be readily deduced. While HCI offers a variety of theoretical frameworks for addressing this (cognitive theory, activity theory, situated action, etc.), the considerable skill, time and effort required to understand and apply such frameworks makes their practice problematic. Additionally, making such analyses relevant to the software design is not obvious (Rogers, 2004). This has given rise to the nascent field of HCI4D concerned with adapting HCI practice to ICT4D contexts (Ho et al., 2009).

For our purposes, we draw on *Communications Theory* and Ong's theories in particular (Ong, 2002) to look at the social phenomena surrounding communications at the individual and community level. Information, seen as messages that have meaning within a given cultural context, circulate among members of a society, and the means of communication together with the messages in circulation constitute the mindset and shared system of meaning within that society, referred to by Innis as the "cultural ecology" (Babe, 2000). According to Ong, culture is a dynamic process, positioned along a continuum between "orality and literacy", with the mode of communication conditioning how people accumulate, preserve, and share knowledge, and ultimately how they think and structure society (Ong, 2002; Couch, 1996; Babe, 2000). Given that literacy (or its lack) is a major distinguishing characteristic of the targeted user populations, these theories are particularly suitable for examining cultural differences with respect to communications as a means of effecting change in a society. They also highlight the cultural differences that exist between the ICT4D designers, end-users, and technology itself. Moreover, the characteristics that emerge can more readily be mapped into operational project constraints and requirements.

Ong contends that an oral culture is by nature traditional, conservative and situational. Traditional knowledge must be carefully conserved as otherwise, once forgotten or distorted, it is permanently lost. In the absence of written records knowledge is embedded in the stories and practices shared by a community. These are preserved in communal memory which is continuously refreshed by constant re-enactment. Oral knowledge can only be transmitted through direct contact among community members. People must experience these stories and practices at first hand on a recurring basis if they wish to learn and recall them. In this way knowledge manifests itself as concrete experience embedded within the social fabric of daily life. By necessity such a culture is conservative, favouring continuity over experiment and radical change. Here the collective has precedence over the individual, as it is the collective that embodies the shared experience that constitutes the pool of knowledge available to the community. At the same time, this pool of knowledge evolves adaptively, as what is no longer relevant gradually passes from usage and is eventually forgotten.

In contrast, in a literate culture knowledge can be permanently recorded. Society is free to experiment and innovate as the original information can always be retrieved if the experiments fail. As noted by Havelock (1963), such societies by nature engender individualism, speculation, innovation and change. When knowledge is recorded, direct contact is no longer essential as information can be perused in asynchronous privacy. Reading and writing are in themselves solitary activities that engender introspection. This introduces an objective distance between the author and audience, allowing readers to form their own opinions uninfluenced by live contact. In the absence of a shared environment, the context must be described with analytical precision, and abstract concepts are used to synthesise the knowledge embedded in concrete, day-to-day life experience. An analytic viewpoint is more conducive to reflection and speculation, opening the doorway for experimentation, which when successful generates change. These processes underlie the scientific method whereby abstract knowledge is separated from experience and then reapplied to new situations. At the same time, because recorded knowledge is relatively static, when change occurs it is often disruptive.

Ong's original terms of 'oral' and 'literate' to distinguish these two worldviews can be somewhat misleading as what they refer to is not the basic ability to read or write, but rather the extent to which a society has interiorised writing in its thought processes and the value it places on written as opposed to interpersonal sources of information. To avoid confusion in this regard, we henceforth refer to them respectively as *"experiential"* (i.e. grounded in a community's world experience) and *"analytic"* (i.e. derived from analysis and theorising). These two should not be viewed as a dichotomy as in fact they manifest along a continuum and are in constant flux, with diverse influences affecting different aspects of an individual's life.

4.6 Research methods

Both quantitative and qualitative research approaches are methods of scientific inquiry that rely on empirical observation to systematically investigate phenomena in order to develop knowledge. Their fundamental difference resides in their underlying assumptions about the nature of the world and knowledge. Whereas quantitative research views scientific knowledge as objective fact (or truth), describing a single reality that has quantitatively measurable properties, qualitative research considers reality as socially constructed with multifaceted meaning (Myers, 1997). While previously their differences were considered irreconcilable and the source of much debate (and even clashes), they are now viewed as part of a continuum. Given that *"all quantitative data is based upon qualitative judgments; and all qualitative data can be described and manipulated numerically"* (emphasis original, Trochim, 2006), the debate becomes one of culture and tradition. Their alternative views provide complementary perspectives, and mixed-method approaches are used to enrich research results and make them more reliable (Casebeer & Verhoef 1997, Creswell 2003, Johnson & Onwuegbuzie 2004, Mingers 2001). The choice of research method is largely determined by the nature of the question, the purpose to which the answer will be put, and the expectations of the audience. Below, we provide a brief overview and list their respective strengths and weaknesses.

Quantitative methods are associated with a positivist paradigm that considers scientific knowledge to be objective truth describing a single reality with measurable properties independent of the observer. They originated in the natural sciences and until recently dominated information systems research (Mingers, 2001). They generally apply a deductive approach to test preconceived theory by providing evidence that supports or refutes a specified hypothesis (hypothesis testing). Emphasis is placed on explaining phenomena in an unbiased and value-free way, by demonstrating causal relationships established through statistical analysis of data collected under controlled conditions using standardised, objective measurements expressed in numerical form. The two strategies of inquiry are experiments and surveys. The data collected is predetermined and fixed, and questions are close-ended. The value of a quantitative approach resides in the predictive ability of its results, whose validity largely depends upon the methodological rigour of the study. Key factors are the operationalisation of the theory, the goodness of the sample (i.e. its representativeness), the reliability of the measurement device (i.e. consistency of the results it produces), the validity of measurements (i.e. the extent to which a measurement measures what it was intended to measure), and the statistical validity of the conclusions drawn from the data. The positivist paradigm is central to the scientific method whereby knowledge is discovered by collecting measurements under controlled conditions and applying statistical techniques to produce replicable findings that can be generalised to a broader population.

The recognised strengths of quantitative methods include (Johnson & Onwuegbuzie, 2004), their utility for validating already constructed theories and providing statistics on large populations. Also, data collection and analysis are relatively fast and provide precise numeric results that are independent of the researcher. Their weaknesses include the need to establish theory a priori, with the risk that the theory or constructs do not correspond to local reality, and thus the research misses out on relevant phenomena. This can have significant consequences on results, leading to spurious correlations and conflicting interpretations because of missing factors in the model (Pearl, 2000). Comparable problems arise if measurements are incompatible with local understanding, generating misleading responses. Finally, the knowledge obtained may be too abstract or general to be directly applicable to specific situations.

Qualitative methods are generally associated with an interpretivist paradigm that considers reality as socially constructed, and thus subjective. They arose in the social sciences to study social and cultural phenomena by applying an inductive approach to discover meaning behind naturally observed phenomena (theory discovery). Here emphasis is placed on understanding phenomena by discovering the patterns, processes and meanings that underlie observations collected in natural settings. Strategies of inquiry include case studies, ethnography, narrative (history), grounded theory (used to derive a general, abstract theory "grounded" in the views of individuals) and action research (which stresses a participatory approach to problem identification and intervention for change). The data collected is open-ended and can take a wide variety of forms, from observations and interviews to physical artefacts. Qualitative approaches are valued for the detailed, in-depth understanding they provide, and are particularly useful for exploring the how and why behind phenomena. While the sample is important, the validity of results largely depends on the skill and rigour of the analyst, and is generally assessed in terms of "credibility" or "trustworthiness" (Creswell, 2003). This is enhanced by factors such as sufficient observations, awareness of bias, acknowledging discrepant information, an audit trail tracking the emergence of findings from observations. and "rich. thick description". Because of the interpretive nature of the analysis, results are non-replicable in the quantitative sense.

Among their known strengths (Johnson & Onwuegbuzie, 2004), qualitative methods are particularly useful for studying poorly understood and/or complex phenomena, dynamic processes and personal experience. Participants are free to express themselves in their own terms, thus overcoming the confounds arising from omission and misunderstanding with quantitative methods. Studies are readily adaptable to local conditions, and the focus of study can easily be shifted in response to observations. Their weaknesses include the inability to generalise findings, the difficulty of testing hypotheses and theories, the time it takes to collect and analyse data, and their susceptibility to personal idiosyncrasies and bias.

Chapter 5: Proposed Models and Methods

The methodology we propose in this thesis draws on the concept of storytelling and the theories of experiential and analytic culture to expand the scope of requirements elicitation in an ICT4D context. We augment the standard requirements engineering process by applying *Structured Digital Storytelling* (SDS) to elicit needs directly from end-users, and apply a conceptual model of experiential culture to interpret these needs and additional constraints arising from the broader sociocultural context. In elaborating the needs, we incorporate properties that a system should possess to be successful in an ICT4D context, based on a model of technology use within a rural context.

This chapter presents the models and methods developed in the course of elaborating the SDS methodology. We first present our model of technology use in a rural society (Pitula & Radahakrishan, 2007a), and the set of properties that an ICT4D project should possess to "fit" within a given rural context (Pitula & Radhakrishnan, 2007b). Next, we describe our model of experiential culture for characterising a society's culture based on the inherent differences between an oral and literate society (Pitula et al., 2010). We follow by describing the technique of *Structured Digital Storytelling* (Pitula et al., in press), and compare it to the techniques conventionally used for requirements elicitation. The description of the SDS methodology itself is detailed in the chapter that follows.

5.1 Model of technology use in a rural society

We propose a conceptual model that lays out the key factors involved in designing a technology to bring sustainable, measurable benefits to a rural community (Pitula, 2007). This model, depicted in Figure 8, is based on Maslow's Theory whereby needs motivate human behaviour (Huitt, 2004). According to our model, the rural environment in which a community is embedded largely shapes that community's socioeconomic activity, which in turn determines that community's needs. A community is composed of individuals interconnected in various ways. Needs motivate an individual to identify goals whose achievement will result in a quantifiable or qualifiable gain, which is the motivating factor for undertaking that activity. Achieving these goals requires both knowledge and action. Actions may change the individual's situation and thus the community, in turn reshaping the environment in which the community is located. Acquiring knowledge and acting upon it both require a set of three components: skills, resources and tools. This set can be divided into two disjoint subsets of ICT specific and non-ICT specific components. A software project can only affect the former. While the latter is beyond the scope of a software project, it will be essential in making a difference to the targeted society. Consequently, a software project must be in harmony and support the latter if it is to have an effective impact.

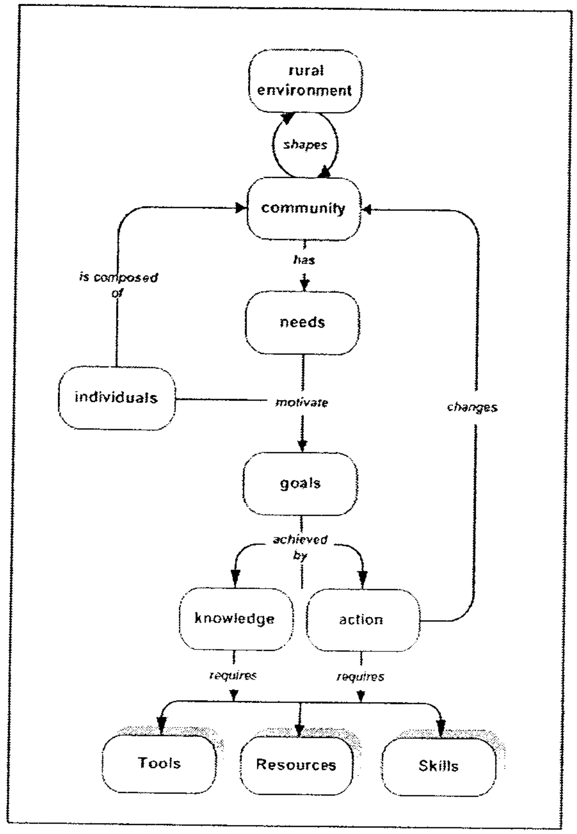

Figure 8. Conceptual model of technology with respect to a rural society

Within this model a sustainable cycle is achieved by selecting goals that balance social and economic benefits. The cycle consists of needs stimulating the discovery of relevant knowledge that leads to actions producing some benefits. As the community's situation improves, its needs evolve creating more "wants" that stimulate the discovery of more relevant knowledge in an on-going cycle. This process is illustrated in Figure 7. However, given a goal, there is rarely a simple, one-to-one relationship between the knowledge and action required to achieve that goal. Often, realising a high-level goal will involve a set of knowledge and actions that interact iteratively until a satisfactory state is reached. Drawing an example from the e-Choupal project (Tongia & Subrahmanian, 2006), a farmer may discover that he can obtain a better grain price by selling directly to the grain company as opposed to a middleman. However, in order to benefit from that knowledge, the farmer must first find out where to go, whom to talk to, how to transport the grain, and any additional costs involved along with the price difference before making a decision, let alone acting upon it.

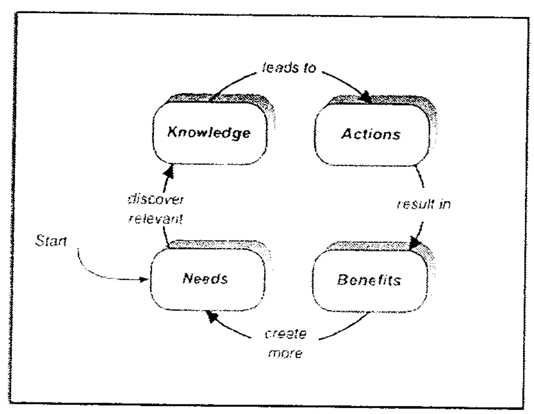

Figure 7. Sustainable cycle

Our focus is on the ICT specific skills, resources, and tools and ensuring that they support a community in acquiring the knowledge and developing the skills it needs to achieve its goals and improve its situation. Towards this end we associate certain attributes with each of the nodes in Figure 8, based on our characterisation of a rural environment and the barriers to the use and acceptance of technology. For a given project these attributes are assigned specific values characterising the targeted community and technology (Pitula & Radhakrishnan, 2007a). These attributes are listed in the following table.

Table 2. Attributes associated with conceptual model of technology with respect to a rural society

Rural environment	Community
▪ population density ▪ transportation & communications networks ▪ remoteness with respect to urban centres and other communities ▪ climatic conditions affecting technology	▪ cost of transportation & communications ▪ availability & cost of electricity, phone lines, high-speed internet connections ▪ current economic & social activities ▪ organisations (public, private, non-profit, community, social) ▪ services (schools, health clinics, banking, government, etc.) ▪ funding sources
Individual	
▪ household income ▪ livelihood ▪ reading and writing skills ▪ languages ▪ other skills	
Need	**Goal**
▪ function of the community & individual situation	▪ identified by the community & individual to fulfill a need

ICT tools	ICT resources	ICT skills
▪ devices, peripherals, etc. ▪ power sources ▪ Internet connectivity ▪ parts, maintenance and upgrades	▪ content ▪ applications ▪ training ▪ peer support ▪ maintenance and updates	▪ operate ICT tools ▪ run applications ▪ access and create content ▪ technical and administrative support

Drawing on our characterisation of ICT4D projects and the social and cultural aspects of technology acceptance and use described previously, we identify a set of properties (or non-functional requirements) that an ICT4D project should possess in order to increase its likelihood of success. We refer to these as ICT4D properties and describe them in Table 3. Each property is associated with a set of conditions or constraints to satisfy within the local context. These properties are applied in identifying and elaborating the set of ICT specific tools, resources and skills that comprise a given project to ensure that they are appropriate for the targeted rural society. As with conventional software properties, certain properties may have hard binary values whereas others may have to be simply "satisficed". The ICT4D quality model can be applied either to assist in elaborating a project's requirements or to evaluate an existing project (Pitula & Radhakrishnan, 2007b). Note that the ICT4D model is applied in conjunction with conventional quality models identifying standard software properties such as usability, reliability, and so forth.

Table 3. ICT4D quality model

Property	**Characterisation**
Feasible	The ICT tools, resources and skills can be made available on site, given the prevailing conditions and constraints. Factors to consider include: • transportable to or available on site • operate effectively given climatic conditions and available infrastructure • secure housing • on site administration and maintenance • timely parts and updates • special language and skill requirements • development of local skills
Affordable	The cost/benefit is sound given the economic situation of individuals and the overall community in terms of start-up and on-going costs for acquiring, operating, maintaining, using and otherwise benefiting from the technology. Costs to consider include: • acquisition costs (cost of equipment, installation and connection) • on-going operating costs (power, Internet connectivity, rental) • software licenses and subscriptions • training materials and programs • technical support and consultation fees • administration, maintenance and overhead • usage fees to individuals and the community
Accessible	The ICT tools, resources and skills are accessible to a critical mass of the population, including groups that might be marginalised for political, economic or social reasons. • physical access and proximity • required languages and skills • non-discriminatory in terms of gender, age, social class, ethnicity
Relevant	The ICT tools, resources and skills are significant and relevant with respect to the needs and goals of individuals and the community. • current needs and goals • current economic activities • existing knowledge and skill set
Trustworthy	The ICT tools, resources and skills are available, reliable, accurate, and inspire confidence in users. Depending on the project, this may encompass security, privacy and safety concerns. • available when needed • outputs are accurate and predictable • no undesirable or dangerous side-effects (physical, social, political)

Beneficial	Contribute to a positive and measurable outcome in line with the goals of individual users and the community as a whole: • addresses high priority local needs • produces observable and measurable improvements • does not produce negative side-effects
Sustainable	Generate an evolving stream of cost-effective benefits to users and the community as a whole: • benefits are cost-effective with respect to operating costs (both short and long term) • repeated usage brings on-going benefits • can address local emergent needs • scalable as user population grows • locally maintainable
Supports a community of practice	Develop and sustain a community of practice through means such as: • learning 'space and place' • communal participation and peer support • use by experienced masters observable • novices can gain mastery and potentially become 'champions' • promotes both active and passive participation (i.e. creation and consumption) • promotes relationships with funders and policy makers
Culturally appropriate	In line with local practices, customs, values and beliefs. Factors to consider include: • nature of information and how it circulates • current practices and underlying goals and values • decision criteria and constraints guiding choices • relations with local institutions and organisations • existing social structures • social endorsement

Although certain properties may appear self-evident, their explicit assertion with respect to the target context avoids the inadvertent introduction of unstated assumptions regarding local conditions. As with non-functional requirements in general, ICT4D properties are interrelated and their operationalisation can lead to positive or negative side-effects. For example, while a satellite connection might make Internet connectivity feasible, its high-cost will have a negative impact on affordability, with consequences on accessibility. By asserting these properties with respect to the set of ICT tools, resources and skills required for a given project, we ensure that consideration is given to critical contextual factors that might otherwise be overlooked.

5.2 Model of Experiential Culture

To determine the cultural appropriateness of some ICT technology, it is necessary to characterise the culture of that society. The conceptual model we apply for analysing the sociocultural factors with respect to technology is based on the cultural differences that arise between an *experiential* and *analytic* society. These two worldviews are associated with very different and even dissonant characteristics in the societies that embrace them. The table below, derived from (Ong, 2002; Couch, 1996), summarises some of the characteristic attitudes, traits, and tendencies.

Table 4. Traits and tendencies of experiential versus analytic cultures

Experiential culture	Analytic culture
traditional	experimental, seek change
conservative	innovative, seek novelty
knowledge expressed through human action	knowledge expressed abstractly
situational thinking with concepts drawn from concrete experience in operational frames of reference	analytic thinking with abstract concepts organised in logical categories and lists
shared, collective experience	individual, subjective experience
participatory, emotional	detached, objective
high-context communication (context construed from shared environment)	low-context communication (context explicitly stated)
situated learning	theoretical learning
thoughts expressed nonlinearly in additive grammatical structures using formulaic expressions, copious	thoughts expressed linearly as "spatially" organised arguments, using subordinative structures, analytically sparse and precise
live in the immediate present with time fluid and flexible (Hall's "polychronic" perception of time)	live in computed time managed linearly (Halls' "monochronic" perception of time)
collective has precedence	individual has precedence
social norms enforced by shame with respect to the collective	social norms enforced by an individual's guilt with respect to laws

A society's worldview has profound implications on how people manage knowledge. In an experiential culture, knowledge is accumulated through direct experience, by doing, observing and listening. Knowledge is preserved in memory of both the individual and group, with this memory reinforced through constant repetition. Finally, knowledge is shared through its enactment or by telling. This is depicted in Figure 9. All these activities involve high degrees of interpersonal contact, suffusing knowledge with emotion, empathy and participation in a shared identity. Consequently, in an experiential society the value of some "knowledge" is interlinked with the quality of the human relationship as

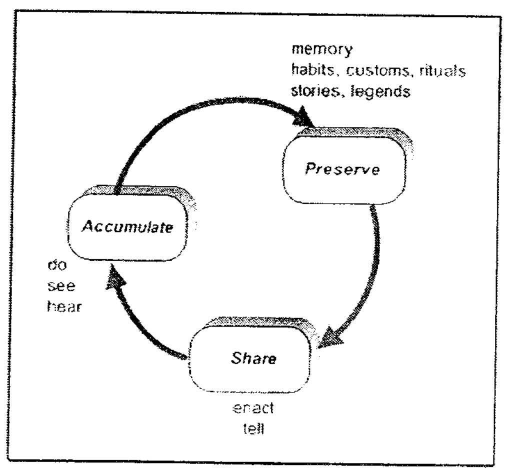

Figure 9. Knowledge management in an experiential culture

opposed to being based solely on its validity, and interpersonal communication is favoured over the impersonal, detached, logical content preferred by analytic societies.

The characteristics of an experiential society introduce certain constraints to bear in mind when considering potential technological interventions and their ability to effect change in a society. The following are among the constraints identified:

1. **averse to disruptive change**

   Given that traditional knowledge is embedded within their practices, people are averse to disruptive changes that threaten the continuity of this knowledge. To offset this any proposed change must be gradual and build on existing practices.

2. **reluctant to experiment**

   Related to an aversion to change is a reluctance to experiment, as this goes against the preservation of traditional knowledge through its constant reenactment. This reluctance can also be offset by evolving existing practices gradually.

3. **knowledge conveyed through concrete experience**

   Knowledge that is embedded in practices manifests itself through human action. Concepts are drawn from concrete experience, and situated in operational frames of reference. Learning is a situated activity in which the novice learns by observing and emulating the expert without reference to underlying principles. Conveying knowledge within such a context is best achieved by a situated, hands-on learning experience that builds on existing knowledge in familiar situations rather than disconnected, theoretical presentations, and by providing observable results as opposed to rational explanations.

4. **not predisposed to abstract thinking**

   Situational thinking is not conducive to abstract analysis or to formulating abstract plans, with concepts expressed in spatially organised and analytically sparse structures. To counteract this, abstract concepts and plans must be presented in terms of concrete experience, situated within an individual's operational frames of reference.

5. **reluctant to act individually**

   Living in a small, tight-knit community bound by a shared, collective experience, people will be reluctant to take a path that sets them apart. This can be offset by promoting group participation in any initiatives.

6. **high-context, personal communication**

The collective experience and shared environment is conducive to high-context communication. By nature such societies are participatory and rich in interpersonal relationships and emotion. Unaccustomed to low-context communication, people are likely to be overwhelmed by highly detailed, analytical information, and have difficulty relating it to their personal context, which remains largely unanalysed. They are apt to be put off by impersonal communication as they will have difficulty relating to it, and consequently distrust it. To offset this, information should be communicated in high-context, and presented by someone with whom they can relate.

To determine the dominant cultural tendencies in a society we propose collecting and analysing stories. The styles and forms of speech as well as the informational content of the stories all contribute to establishing how people relate to the events that they are describing. To characterise the discourse of these two worldviews we draw on the distinction made by Taylor (2008) between "showing" and "telling". With showing, the narrator **effaces** himself from the narrative, giving an impersonal, objective account that conveys a factual nature to the events described. This is the dominant form for expressing scientific knowledge and typical of an analytic society. In contrast, in the telling mode, the narrator **participates** in the events described, imbuing the account with personal emotion and opinion. Drawing on this distinction and the characteristics described in Table 4, experiential narratives can be expected to display the following salient features:

- Speak in concrete terms based on lived experience in operational frames of reference
- Directly implicated in the narrative in which they play a central role; speak in the active first-person (as opposed to detached third-person); convey personal feelings and emotions
- Narratives are high-context with unstated facts construed from the location or activity
- Do not speak of unfamiliar concepts that are outside known frames of reference
- Do not speculate about alternatives, tradeoffs or underlying causes
- View interpersonal interactions in terms of personal relationships (trust, honesty, kindness, fairness, etc.) rather than economic or bureaucratic transactions (rules, principles, legal obligations, etc.)

Given that generally ICT4D end-users and software designers come from opposite ends of the spectrum, it is useful to note certain practical considerations arising from cultural differences between the two worldviews (Ong, 2002):

- People from experiential societies are likely to have difficulty construing the meaning of logically organised artefacts such as lists, tables, charts and diagrams that represent abstract concepts spatially. As these are preferred media for people from analytic societies, it is important for practitioners to reformulate such artefacts into situational frames.
- People from experiential societies more readily accept inconsistencies and contradictions when these make sense in the context in which they occur, in contrast to people from analytic societies who strive to find underlying principles.
- A linear plot line is a construct of an analytic society. In an experiential culture narrative does not necessarily have a chronological ordering but rather consists of a collection of episodes organised thematically and ordered according to the demands of the situation.

5.3 Structured Digital Storytelling (SDS)

Our notion of storytelling as a communications medium for villagers arises from consideration of how the characteristics of an experiential culture affect requirements elicitation. Lacking an operational frame of reference with respect to some hitherto untried (and potentially unknown) technology, villagers are likely to have difficulty relating it to their own experience. Being disinclined to abstract thinking, villagers are unlikely to reflect upon the way they do things, let alone speculate or voice opinions regarding alternative ways that things could be done. Their reluctance to experiment, to differentiate themselves from others and their aversion to disruptive change all act as further deterrents in this regard. Accustomed to high-context communication, they may misconstrue directed questions or provide answers that need to be contextually interpreted. Consequently, their ability to provide meaningful responses using conventional elicitation techniques will be limited. Telling stories, on the other hand, is an activity with which everyone is familiar and additionally, given socially acceptable topics, most people enjoy talking about themselves regardless of their literacy level. Storytelling is by nature a communal activity. Moreover, stories provide useful insights into the users' activities, experiences and social world—the very information sought by requirements analysts.

Our technique of *Structured Digital Storytelling* (SDS) builds on the concept of digital storytelling to which we add a goal-oriented specialisation. Storytelling at its simplest consists of someone telling their personal story on some topic and the narration being recorded. An interviewer may assist to ensure that items of interest are clarified and expanded. Recent digital technologies support the authoring of sophisticated multimedia stories that can be made accessible to a broad audience. *Interactive Voice Response* (IVR) systems offer an alternative approach for collecting user input. IVR applications use structured dialogues to ask a sequence of questions when eliciting information. Our approach combines the two—by adding a multimedia, structured dialogue interface onto digital storytelling technology we can assist people in expressing their information needs through stories which can then be shared in the community. Instead of asking focused questions about their information needs, the villagers' needs are elicited through a series of open-ended questions, short stories, 'what if scenarios' or by hearing their neighbours' views regarding the issues and frustrations they face. The structured dialogue ensures that relevant themes are covered, while hearing stories told by their neighbours will inspire people to tell their own stories.

In an ICT4D context, SDS has a number of advantages over commonly used requirements elicitation techniques such as interviews, focus groups and ethnographic studies (Leffingwell & Wedrig, 2003). The limitations of ethnographic studies for eliciting requirements in an ICT4D context have already been discussed in section 4.5. Below we briefly cover the advantages and disadvantages of interviews and focus groups.

With interviews, facilitators meet with end-users individually to ask them questions regarding their problems and needs. The interviews may be structured, semi-structured, or unstructured. In a structured interview, the questions are pre-defined and all users are asked the same questions whereas with unstructured interviews, the interviewer asks questions on-the-fly. A semi-structured interview combines the two, with the interviewer asking questions from a pre-defined list, and follow-up questions as needed. Interviews have the advantages that both verbal and non-verbal responses can be observed, and users can be probed in-depth with follow-up questions. Among the known disadvantages, tacit knowledge is difficult to elicit and information on the context-of-use is not readily observable. Given the novel nature of the ICT4D context, there is no way of ensuring that all relevant aspects are covered by the questions. Furthermore, because of the socioeconomic differences, end-users may have difficulty answering direct questions or be intimidated by the interviewer. Additionally, conducting individual interviews is time-consuming and requires someone who speaks the local language.

Focus groups are similar to interviews, with the difference that users participate in a group rather than individually. In a focus group, a facilitator presents the group a series of pre-defined questions or topics which the participants then discuss. Along with the advantages and disadvantages of interviews, focus groups have the advantage that the discussion among participants may reveal more requirements and they are less time-consuming than individual interviews. The disadvantages are that participants may feel uncomfortable stating opinions that differ from those of the group, leading to "groupthink". Also certain participants may dominate the discussion, leaving other valid viewpoints unexplored.

A major difference between SDS and the other elicitation techniques is that the narrator is largely left on their own to tell their story in their local language. Among the advantages we foresee with SDS are that storytelling capitalises on the villagers` primary mode of communication. Although the questions provide some general guidance, an interviewer is not present to influence the narration. Thus it is possible to identify problems and needs not initially envisaged and contextual factors that might otherwise be overlooked. Moreover, collecting stories involves fewer resources in terms of facilitators, preparation, and elapsed time, and it does not require facilitators who speak the local language. A disadvantage with respect to the other elicitation techniques is that there is no human being present to provide clarifications, guidance or immediate follow-up on items of interest. Additionally, participants may focus their story on one aspect, leaving other equally relevant aspects unmentioned. Among the challenges to address, storytelling is by nature a social exchange that is enriched by the presence of an audience. As with any technique, consideration must also be given to confidentiality and self-censure, particularly when dealing with sensitive subjects.

Chapter 6: The SDS requirements elicitation methodology

Our methodology, specifically designed for ICT4D projects, applies established software engineering principles such as user-driven requirements, goal oriented analysis, and requirements validation based on traceability to user needs. It augments the standard requirements engineering process by applying *Structured Digital Storytelling* to elicit requirements and contextual information directly from end-users, and a model of experiential culture to identify cultural factors that are not directly observable. Using a goal-based analysis, the outputs of this process are incorporated into the standard RE process to provide a bottom-up view of the potential areas of technology intervention, while the cultural model is applied to identify additional constraints. The ICT4D quality model is applied in parallel to ensure the quality of any intervention with respect to enabling social development. An overview of the augmented RE process is provided in Figure 10, followed by a detailed description of the SDS process integrated within a conventional RE process, with a focus on the elicitation phase, and the requirements specification and validation activities as they relate to the SDS outputs and models produced during elicitation.

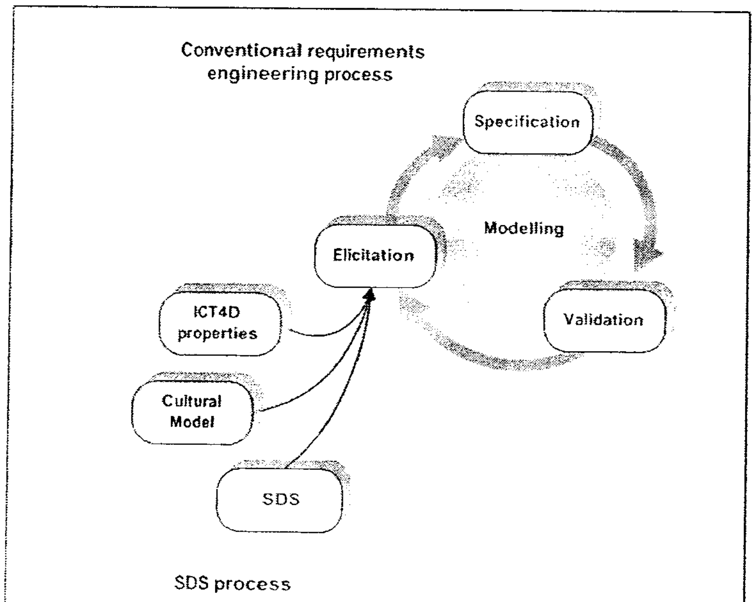

Figure 10. Conventional requirements engineering process augmented by SDS process

6.1 Needs elicitation

Elicitation in the standard RE process encompasses the activities required to understand the problem the proposed system will address, delimit the system boundaries and identify requirements. Modelling plays a key role, with domain, task and goal models common techniques for representing the problem space. The SDS process fits within these elicitation activities to assist in constructing the models based on which the software requirements will be derived. ICT4D projects take place under the auspices of development organisations, funding agencies or technology related businesses, to deliver information or services of value concerning some perceived need in the population. Given this high-level goal, requirements elicitation generally starts with determining what categories of information or services pertaining to that need are critical to the targeted population in prioritised order. These categories correspond to potential areas of intervention for the project, and at an operational level, relate to a community's economic activities (farming, fishing, etc.), or the well-being of its families (health, education, governance, etc.). While external stakeholders' needs can be elicited using conventional techniques, rural users are likely to have difficulty articulating their needs for the reasons already discussed. It is here that we propose using SDS techniques supported by a suitably designed interactive multi-media software tool, to elicit users' needs through stories.

The SDS process consists of collecting oral narratives expressed from an experiential perspective, and transforming them into an analytic representation suitable for identifying requirements. The process involves 4 major steps: (a) identifying themes of interest; (b) collecting stories; (c) processing the stories to extract information; and (d) modelling the extracted information, as illustrated in Figure 11. SDS can be applied at any stage of requirements elicitation, to assist in identifying areas of intervention, to identify and validate high-level goals and constraints, or to elaborate and validate operational goals and conditions. The use of an SDS approach does not exclude the use of other elicitation techniques. In certain cases it may be complementary, assisting stakeholders in validating that they are focusing on the right problem, and that the problem is thoroughly understood in the context in which it occurs.

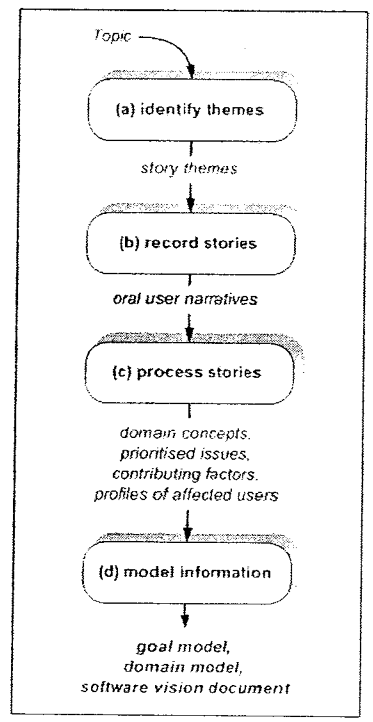

Figure 11. SDS process

6.2 Identifying themes

Given a topic, the SDS process starts with selecting the focus of elicitation. Depending on how well understood the high-level need is, emphasis may be placed on eliciting general contextual information (e.g. farming in general), or alternatively on eliciting detailed information about a particular activity or event (e.g. planting or selling produce). Working with domain experts, themes of interest are identified and prioritised in order to arrive at an optimal number of open-ended questions to produce stories of acceptable length. Assuming 1–5 minutes of narration per theme, 5 themes plus or minus 2, should be reasonable. A theme may be allocated to eliciting demographic information, or alternatively, a short sequence of focused questions may be asked. We refer to the set of themes associated with some topic as a "story".

When selecting themes it is important to consider the experiential nature of the targeted rural societies. While readily able to talk about things they do, people they encounter, and events they experience in the context of concrete, familiar situations, participants may have more difficulty responding to themes relating to abstract categories. For example, "borrowing money to buy seed" is an event well situated in a "planting" context, whereas it becomes a detached concept if associated with the abstract category "financing farming activities". When identifying themes, coarse-grained domain and task models can be used to relate the information sought to the concrete situations in which it occurs. The local vernacular should also be considered when formulating, translating and recording questions. With regards to wording, we emphasise that questions should be open-ended to elicit broad coverage without going into specific techniques for encouraging talkativeness.

6.3 Recording stories

Once the themes have been identified, recording stories is relatively straightforward. The set up will vary depending on available amenities. An application (such as the E-Tool presented in the next section) that plays the questions and records responses can be made available on a suitable device (e.g. laptop or mobile phone) in a location such as a community centre, and villagers invited to use it. As with any such undertaking, a local champion and the acquiescence of local leaders will favour participation. Enlisting respected members of the community to record their own stories as examples for the story library will provide further encouragement, and let participants "hear" how to respond to the themes. Time-wise, the technique takes the time required to tell a story, allowing a reasonable number of stories to be recorded in a few days. Because resources such as interviewers and facilitators are not involved, the application can readily be deployed in a number of villages, increasing the number of the stories recorded and coverage of issues.

6.4 Processing stories

Once the stories have been collected, they are first transcribed and translated before being analysed to determine what concerns and problems participants mention, their importance, and other noteworthy elements. While transcription and translation are relatively straightforward, the analysis requires more skill, to identify and classify the issues, establish what factors are relevant and how they are related, and to determine the cultural tendencies that participants manifest. The various steps involved in processing and analysing the stories are depicted in Figure 12, and described below with examples drawn from the case study.

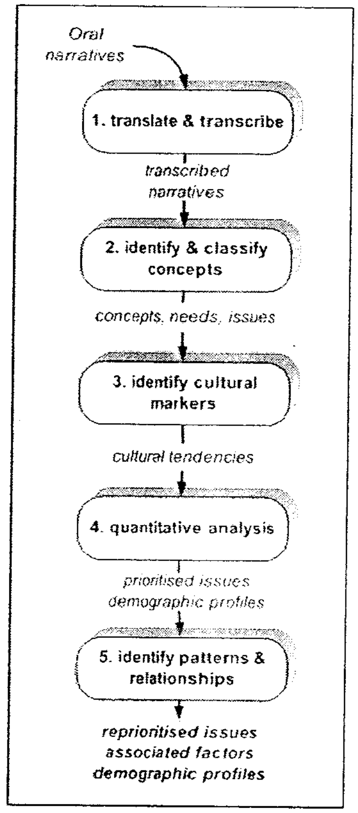

Figure 12. Steps involved in processing stories

Step 1—translation and transcription: Since the stories are narrated in the local language, they are first transcribed and translated into English (or other working language). This is a manual activity that is somewhat labour intensive, but does not require specialised skills beyond knowledge of the local language, English, and basic literacy. Translation can be carried out by people from the same region with appropriate qualifications. The output of this step is the transcribed stories.

Step 2—concept identification and classification: The narratives are examined to extract demographic information and identify significant domain concepts, including problems, issues and concerns as well as additional needs and desirable features. Elements such as activities, actions, agents, objects, events, locations, opinions and attributes are identified and classified, while issues and concerns are flagged and their significance characterised. These are abstracted into concepts for inclusion in the domain model. For example, farmers may speak of lacking a well, the well going dry, the rains being late, crops requiring too much water, etc. All of these narrations are classified under the concept "irrigation" with the problems flagged as "irrigation issues". Relationships are of particular interest, as they are often central to problems. Thus, for example, when farmers speak of "being compelled" or "having no recourse" in the context of selling their produce, these point to an unequal power relationship, while attributes such as "dishonest", "unreliable", etc., indicate trust issues. All the concepts that emerge from the stories are noted, even if mentioned only once, as they might provide a critical insight into some unknown or poorly understood phenomenon, especially when narrations are high-context. This phase produces the set of concepts to include in the domain model, additional needs and features for the goal model, and a summary view of the individual stories, highlighting the subset of concepts contained in each.

Due to its qualitative nature, certain aspects of this analysis, such as identifying and classifying concepts (coding) or extrapolating goals, require the attention of a skilled person. A tagging tool can be developed to assist the manual process of mapping information in the narratives to concepts and developing a classification scheme. When dealing with large amounts of data, semi-automated or automated natural language processing can also be applied. Corpus linguistics has been used to process documents for requirements engineering in a number of problem domains. A general overview of the application of natural language processing to requirements engineering is provided in (Rolland & Proix, 1992). There is also a number of commercially available software packages specifically developed for analysing qualitative data.

Step 3—cultural tendencies: In this step the stories are analysed to identify the cultural tendencies manifested. As we are dealing with translations, the analysis focuses on what participants speak of and the voice used rather than the syntax and vocabulary of the narratives. This analysis draws on the markers from our cultural model to characterise narratives as grounded/theoretical, implicated/detached, high/low-context, and emotional/rational. These characteristics can be associated with an entire story, or with a particular theme or concept, thus allowing for tracking different cultural tendencies within the same story. This step establishes the cultural characteristics present in the stories. Currently, we view this as a manual process that can be conducted in parallel or following step 2, with the assistance of similar tagging tools.

Step 4—quantitative analysis of stories: Assuming a sufficient number of stories have been collected, this step uses the summarised stories produced in step 2 to provide a quantitative overall view of the information. It presents demographic information on the participants and the incidence of the various problems and concerns mentioned, allowing these to be prioritised. Of interest also are the problems which receive few mentions but are critical in nature, as there may be underlying reasons for their omission from the narratives. The output of this step, which ideally is fully automated, is a list of prioritised issues, concerns and omissions that serves to focus the next step.

Step 5—profiles, patterns and relationships: Given a problem, this analysis probes to understand the underlying factors that contribute to it, or alternatively, prevent it from occurring.

Here we look at the demographic profile of the people who mention a problem, other problems they mention, and make comparisons with those who don't, to seek patterns in the data that will permit us to establish relationships between a problem or set of problems and the set of conditions characterising those it affects. For example, while all the farmers mention problems with pests, less complain about the cost of pesticides, and even fewer complain about its availability. A profile analysis reveals that those complaining about cost and availability are the larger, better off landholders who can afford to apply pesticides, and thus are affected by its cost and availability. Probing the differences between these two groups further, we discover that while larger landholders are effusive in their complaints concerning selling their produce (e.g. commissions, cheating, unfair pricing), smaller landholders are largely silent. Seeking a reason for this omission, we discover that farmers who borrow money are compelled to sell to the lender, thus bypassing the open market system for selling produce. Such omitted issues that one would expect to be significant, get particular attention as they may conceal underlying problems warranting further investigation such as subjects that are taboo, a fear of repercussions, or some socially undesirable behaviour (e.g. borrowing money). Because the data is derived from narratives rather than structured surveys, this analysis is considered exploratory, revealing possible underlying factors and relationships without establishing their statistical prevalence or significance. While the interpretation of such patterns is a skilled manual activity, automated data mining techniques can be applied to detect patterns within large bodies of data. We refer readers to the related literature. The output of this step is a reprioritised list of issues along with contributing factors and the demographic profiles of affected users, including their cultural tendencies.

6.5 SDS modelling

Modelling is central to requirements engineering as it provides an abstract representation of the problem space based on which system goals and boundaries are established, and requirements defined. The SDS process contributes by providing inputs for representing and analysing the problem from the users' perspective. Here we draw on Domain models, Influence Diagrams (Howard & Matheson, 2005), Causal Loop Diagrams (Sterman, 2001), and Goal Models (Van Lamsweerde, 2001) to represent and transform the SDS outputs into a representation that serves as a basis for elaborating software requirements. These models are used in a descriptive fashion to assist in developing and communicating the static and dynamic nature of the problems described in the stories, to analyse the potential effects of possible interventions, and to identify specific needs and constraints, leading to high-level functional requirements whereby they are addressed. The SDS modelling process is depicted in Figure 13 and described below.

Step 1—Domain model: SDS modelling starts with constructing a problem domain model with the concepts, needs and issues extracted from the stories (see Figure 20 in the following section). This model serves to identify and organise the concepts, establish their relationships, and classify the associated needs, concerns and issues, all of which are linked back to the specific stories and themes in which they are mentioned. This model serves as a base reference for the other models.

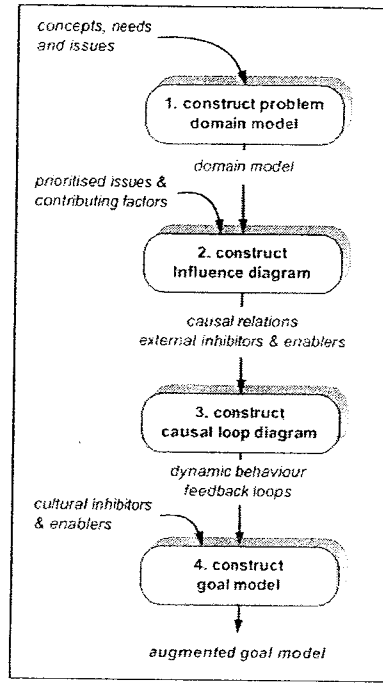

Figure 13. SDS modelling process

Step 2—Influence Diagrams: Given a specific issue and set of contributing factors (from the list of prioritised issues and associated factors identified in step 5 of processing stories), the elements that comprise that issue are modelled as a decision situation. Using influence diagram notation, the problem is modelled in terms of its objectives (i.e. the issue being studied), variables, decisions, and outcomes in order to explicitly represent and reason about the causal relationships that exist among the elements and their influence on each other (see Figure 21). While the term "decision" implies some choice, in an ICT4D context this choice is often dictated by an individual's circumstances which may impose hard constraints on the options available (e.g. farmers who lack funds must borrow money; farmers without storage facilities must sell their produce immediately). These constraints can be tangible things such as the availability of time, money or tools, to more insubstantial factors such as government policies, access to funding, or other incentives that influence a person's behaviour and exert a positive or negative influence on an individual's ability to act. We refer to these real world obstacles or aids as *external* constraints, and introduce the terms *inhibitor* and *enabler* to refer to those that respectively impede or facilitate some course of action.

Modelling the problem as a decision situation brings out the underlying cause-effect structure in terms of the inhibitors and enablers that influence decisions and their consequences on outcomes, with different circumstances giving rise to different paths. The dynamics of these paths can then be modelled as causal loop diagrams, with the influence diagram serving as the base reference. Influence diagrams are also useful for deliberating about goals as they can readily be mapped into high-level goal models, with the objective to optimise mapped to the high-level goal, and the contributing factors mapped to subgoals, as illustrated in Figure 23. Influence diagrams can be created to model different issues at different levels of abstraction, depending on the complexity of the problem being studied. The link to the stories is retained by relating the elements in the influence diagram to the concepts and relations in the domain model.

Step 3—Causal Loop Diagrams (CLD): are particularly appropriate for modelling the different paths through the influence diagram and for explicitly representing the various inhibitors and enablers and their positive or negative influence on outcomes. These paths can be modelled as linear cause-effect chains (or open loops). However, as the problems being studied are dynamic, real-world situations, they often contain feedback loops at the core of the issues mentioned. Modelling the problem using CLD permits us to reason about the system in terms of its dynamic behaviour and to identify problematic feedback loops that trigger or exacerbate the situation. For example, lacking funds to buy supplies, farmers are obliged to borrow with the condition that they sell their produce to the lender at below market price. Consequently they have less income, which together with the interest charges reduces their available funds, obliging them to borrow again (see Figure 22). Such negative feedback loops constitute standalone issues in themselves, while positive loops can have a reinforcing influence, leading to the identification of inhibitors and enablers at the systems level. Considering possible interventions with respect to such loops allows us to evaluate the potential impact of these interventions as well as their side-effects. While CLD models can be created directly, deriving them from the related influence diagram ensures that they are grounded in the experience described in the stories. Here again, CLD models can focus on different issues at different levels of abstraction, while the link to the stories is retained by maintaining links to the domain model.

Step 4—Goal Models: The detailed goal model is the representation that synthesises the information derived from the SDS analysis in a form suitable for defining software requirements. This model is developed incrementally by starting with a high-level goal (from the prioritised list of concerns), and then successively decomposing and refining it until a set of operational requirements by which it can be met is attained. To develop the detailed goal model we draw on the other models. Given an issue, a quick examination of the related influence diagram shows its cause-effect structure. By mapping the underlying elements to subgoals, the primary issue can be addressed, producing a preliminary high-level goal model for that issue, with a one-to-one mapping to the related influence diagram. To decompose it further, we now consider the issues associated with the concepts related to each subgoal (e.g. to improve seed supply options we examine the concerns associated with seed documented on the domain model) and together with input from domain experts, reformulate these as more detailed subgoals. The resulting high-level goal model provides an overview of possible interventions with respect to the primary issue (see Figure 23).

In considering what interventions to pursue, we consult the causal loop diagram to assess their potential impact and side effects over the short and long term as well as the inhibitors and enablers present. Whereas some of the farmers' concerns will be addressed, others will not and still others are innate (e.g. a lack of funds or distrust of suppliers). Concerns that will exert a negative influence on an intervention's use are classified as *external inhibitors*, and added to the list of inhibitors identified in the course of the analysis. These include any system level inhibitors corresponding to negative feedback loops on the related causal loop model. Before proceeding, we introduce the notion of *cultural* constraints. The same as external constraints, cultural attitudes can exert a positive or negative influence on behaviour, which we respectively call *cultural inhibitors* and *cultural enablers*. Both external and cultural inhibitors can be addressed by introducing appropriate enablers (or subgoals) to cancel them. The set of enablers associated with some goal correspond to non-functional requirements pertaining to that intervention.

Inhibitors and enablers associated with some goal can be captured using the notation presented in (Chung et al., 2000). With this notation, interdependencies between goals are indicated by arcs labelled '+' or '-' to show the positive or negative influence they proffer as well as a qualitative assessment of their weight. These goals are termed "soft-goals" as there is no clear way of measuring if they are satisfied; instead they are considered "satisficed" if they can be realised within acceptable limits.

Once a suitable intervention has been identified, the corresponding subgoal is refined in a detailed goal model. This is accomplished by introducing the pertinent external and cultural inhibitors and enablers (see Figure 24). We next consider alternative ways in which the enablers can be realised. For example, a possible way for achieving "observable results" with regards to seed suppliers is to show videos of crops. This would be captured as an operational goal and correspond to a high-level system feature or functional requirement (see Figure 25). The set of system features supporting some high-level goal may vary significantly depending on the particular subset of goals selected.

As the set of potential system features emerges, it is evaluated with respect to the ICT4D quality model. Here the realisation of properties such as feasibility or accessibility may introduce additional conditions and constraints, or eliminate contenders outright. For example, while 'viewing videos' might be considered a possible feature, the inability to transfer videos on location because of e.g. poor connectivity, would make that solution unworkable. Alternatively, the optimisation of certain properties might lead to new goals or additional functionality and refinements. For example, supporting a community of practice might lead to introducing the goal of providing a 'learning space' within the system, or the video feature might be refined to support the creation of videos as well as viewing. Thus the operationalisation of non-functional requirements with respect to system goals results in high-level functional requirements (or system features) at the leaves. These constitute the high-level software requirements derived from the needs and constraints expressed in the users' stories, documented in the models and traceable back to the original stories. This modelling activity can be facilitated by suitable automated tools, like those developed to support diverse semi-formal modelling approaches in the goal-oriented requirements engineering literature (Matulevičius et al., 2006).

6.6 SDS outputs

The complete set of artefacts produced by the SDS process consists of a detailed goal-model, supplemented by a domain model, influence diagram, high-level goal model and causal loop diagram as well as a set of non-functional requirements related to the external and cultural inhibitors and enablers identified during the analysis. These constitute the input to the conventional RE process. Their usage depends on the stage of elicitation.

For example, in the early stages of RE, the domain model provides a snapshot of the users' major concerns. Supplemented by influence diagrams and high-level goal models, this subset can be used to set the project vision and estimate its viability and social impact based on the number of users potentially affected. Working together with domain experts and other stakeholders, the problem is analysed and potential solutions are evaluated. The list of prioritised concerns, as perceived by the intended users, serves to focus and potentially redefine high-level project goals, while a profile of affected users helps establish who might benefit from a potential solution and who won't, with the possibility of introducing additional goals to address the reasons for their exclusion. The more detailed causal loop diagram explicitly represents the external inhibitors that need to be dealt with, including problematic feedback loops, while the users' cultural attitudes may introduce yet more constraints regarding what constitutes an acceptable solution. All of these may introduce additional needs and constraints, leading to possible revisions of the estimate and vision.

Once the high-level vision has been set, the detailed goal model with its high-level functional requirements synthesises the analysis results and is the primary input to requirements specification. During the RE specification stage, the SDS requirements are integrated with other system requirements and refined into "well-formed" technical requirements. At this stage, the complete set of SDS non-functional requirements (inhibitors and enablers) can be incorporated, to ensure that new requirements do not conflict with contextual constraints. The other SDS models may also be consulted, particularly when tradeoffs need to be made. Once the technical specification is complete, the SDS goal models can be used in validating the consistency and completeness of design, while design quality can be validated by assessing how well the system's functionality fulfills the users' actual needs and constraints. The goal models can also be used to define metrics for measuring a project's success based on its social impact.

6.7 Requirements Specification

During *specification*, the needs, goals, constraints and high-level features are transformed into a complete set of well-formed software requirements. Here, the detailed goal model of the previous phase serves as primary input for a traditional requirements engineering process. Using a conventional goal-based analysis that starts with the high-level goals identifying the purpose of the system, these are successively decomposed and refined until a set of technical requirements by which these goals can be met is attained. Proceeding in a top-down manner, and integrating statements from the various sources (user goal model, other stakeholders, HCI analysis, operating conditions, business model, software quality properties, etc.), goals are elaborated and analysed, negative and positive interdependencies identified and tradeoffs negotiated. Here, analysts are expected to work closely with domain experts to reconcile differing goals and clarify unstated assumptions. Throughout this process, the information gathered during the elicitation phase serves to prioritise goals, expose obstacles and identify additional goals and constraints to satisfy. The goals resulting from this analysis are refined until a single set of well-formed technical requirements is attained, documented in the final *software requirements specification* (SRS). This specification unambiguously depicts the set of functional and non-functional requirements for the envisioned software product against which its success will be measured.

6.8 Requirements validation

During validation, the SRS is reviewed to ensure its quality of conformance (i.e. the specified requirements address the stated user needs) and quality of design (i.e. the specification fulfills the users' actual needs and expectations). When reviewing non-formal specifications, a goal-based analysis offers many advantages (Van Lamsweerde, 2001). The conformance and completeness of a specification with respect to a set of goals can be established by ensuring that all goals within that set can be achieved, and that all requirements are related to at least one goal within the set. Consistency is ascertained by ensuring that all interdependent goals are "satisficed" with respect to each other. When it comes to validating the quality of design, while some stakeholders might be able to review an SRS, many are overwhelmed by the technical detail, and end-users are customarily excluded. The different levels of abstraction present in a goal-model permits stakeholders to view a specification at a level of detail that they can comprehend. These views are also useful for exploring alternatives, validating choices and detecting and resolving conflicts. Additionally, by maintaining traceability links between the stories, goals, and requirements, any requirement can be traced back to the specific stories in which the need was expressed, thus validating the design with respect to the needs of the intended users. These links also provide the justification and rationale for including any requirement, making it easier to manage requirements as they evolve over the lifetime of a project. Here the relative stability of goals as compared to the wide variability in the set of system features by which these goals can be met makes a goal model particularly useful.

6.9 Project success metrics

Once the SRS has been validated, the related goal model serves as the basis for defining a measurement framework for assessing the project's success in terms of its high-level social development goals, as opposed to basing the assessment solely on its technical success. Drawing on the 5-level measurement framework proposed by Potts Steves and Scholtz (2005), the high-level social development goals are formulated as precise system goals, and each system goal is refined into a set of evaluation objectives. These in turn are refined into conceptual metrics and measures, and the conceptual measures are subsequently defined in implementation specific terms. The result is a precise and pertinent definition of the measurements whereby the project's success will be evaluated from a social development perspective, for inclusion in the project evaluation plan. This measurement framework can be applied to conduct assessments both immediately after a project has been deployed as part of the project's acceptance criteria, and after it has been in the field sufficient time for any social impact to have materialised.

Chapter 7: E-Tool and Its Applications

To assist in validating the SDS elicitation technique, we have developed a prototype application called the E-Tool. The E-Tool is designed to be a self-contained application for eliciting the needs of rural villagers, expressed as narrations in a story like fashion with minimal intervention from outside support staff. Given that the E-Tool is intended for use by a rural population, its design incorporates many of the principles embodied by our ICT4D quality model and model of experiential culture. In this chapter we first present the prototype E-Tool application and the considerations that went into its design. We describe the capabilities of the initial prototype, and propose enhancements pertaining to its scalable deployment. We follow with a projected future version of the E-Tool, intended for a mobile phone platform. We conclude this chapter with some potential applications of the E-Tool, both as an elicitation tool and by extensions to the E-Tool concept itself.

7.1 E-Tool prototype

Our aim in developing the E-Tool prototype is to provide (a) a proof-of-concept for validating our hypothesis that stories are an effective means for collecting information from rural people with (b) minimal or no intervention of facilitators. Along with this primary purpose, we also envisage the E-Tool as a means for villagers to express themselves and to share information regarding their activities. Thus the E-Tool is intended to serve the dual purpose of (i) collecting data in the context of gathering software requirements, and (ii) sharing information in a village setting.

The E-Tool is designed as a stand-alone application for recording and playing back stories in digital audio format. The prototype runs on a portable laptop computer equipped with an inexpensive handheld microphone and built-in speaker, and provides (1) an introductory video explaining the reason for collecting the stories as well as an overview of how to use the E-Tool, (2) a story library where the stories are stored and villagers can listen to any of them, and (3) an interview feature which guides users through a series of questions, letting them tell their own story. Below we first describe our design considerations before presenting the tool itself.

7.1.1 Design considerations

In designing the E-Tool, we drew on the principles expressed in our ICT4D quality model and on our model of experiential culture in particular, to make the E-Tool appropriate (i.e. acceptable and usable) for the widest user base possible within a rural environment. Drawing on our characterisation of the populations targeted by ICT4D projects, we assume the following profile for the E-Tool's user base:

- reside in a rural location
- users of all ages, from school age children to the elderly
- both male and female users
- from a range of socioeconomic backgrounds, from the most disadvantaged to relatively well-off
- a range of education levels, from limited or no schooling to post-secondary, with corresponding literacy levels
- speak primarily local languages
- unfamiliar with computers, but familiar with mobile phones, radios and televisions (even though they may not possess them)

From this profile, low literacy emerges as one of the fundamental user constraints to address, compounded by the use of local languages and unfamiliarity with computers. Given their low literacy, we assume that many users will manifest the characteristics associated with an experiential culture, namely:

- averse to disruptive change
- reluctant to experiment
- knowledge conveyed through concrete experience
- not predisposed to abstract thinking
- reluctant to act individually
- high-context personal communication

To address the constraints related to these cultural traits, we frame the E-Tool within a storytelling metaphor. Storytelling is a familiar and enjoyable communal activity in which people can participate by contributing their own stories. To emphasise this communal aspect, the stories recorded with the E-Tool are saved to a story library where users can go to see and hear the stories of other community members. The library also contains some sample stories, recorded by selected individuals such as local leaders and other people to whom users can relate. These samples are created prior to starting collection in order to ensure that the library is not empty when the general population accesses the tool. The sample stories also provide concrete examples of what users are expected to tell. The communal aspect of the E-Tool is further enhanced by encouraging group participation when telling stories (even though a given story is narrated individually). To make the actual narration of a story easy (and encourage coverage of all topics), a story is broken down into a short sequence of themes or topics on which users are prompted to speak, with the complete "story" constructed from the individual narrations. Information regarding the application's purpose and operation is conveyed through a video.

By presenting itself as a storytelling medium, inviting users to participate and demonstrating how to do so, the E-Tool fits within the constraints of an experiential culture:

- It builds on existing practices (storytelling) that are simply transferred to a new medium;
- It conveys knowledge through concrete experience (examples, observing other users, the video);
- Users do not have to think in order to tell their story (they simply respond to prompts);
- It promotes participation in a storytelling community (story library and group participation), and
- It operates within a high-context mode of communication (users are free to express themselves as they wish).

Applying our ICT4D quality model, we consider additional properties to satisfy. As this is a proof-of-concept, we focus on minimal requirements with respect to needs elicitation in an ICT4D project context, leaving considerations pertaining to the tool's scalability and other improvements for future enhancements.

- **Feasible**: a laptop computer entrusted to a technician or local individual with access to electricity for recharging the battery provides a feasible solution with regards to making the application locally available and operable.
- **Affordable**: as costs are covered by the stakeholders undertaking the project, there is no cost to users or the community.
- **Accessible**: the tool itself does not impose any restrictions on access. E-Tool prompts are in the local language, introducing the need to support the recording and installation of prompts in different languages. Access by the local population mainly depends on the arrangements made for collecting stories.
- **Relevant**: the application is made relevant by selecting significant topics relating to the users' daily activities. A video explaining the purpose of collecting stories provides additional motivation for using the application.

- **Trustworthy**: the reliability of the tool is assured by the technician entrusted with the laptop. When telling stories on socially acceptable topics, confidentiality is not a concern. However, it will be a concern if collecting information on sensitive topics.
- **Beneficial**: in the context of gathering requirements, the tool does not provide the community with any direct and immediate benefit. However indirect benefits include providing input regarding some intended project, and making community members feel that they are participating and having a say in the process
- **Sustainable**: the proof of concept does not address sustainability beyond the ability to change questions and to translate prompts into different languages.
- **Supports a community of practice**: the library and sample stories provide a learning 'space and place' where users can first observe before trying the application themselves, while the video provides additional information. Group participation is a source of peer support while the sample stories let users observe "masters". When recording, the ability to listen and edit what has just been recorded allows users to make corrections.
- **Culturally appropriate**: as explained above, the application fits the traits of an experiential culture and is culturally appropriate for the reasons previously given. Additionally, the application is suitable for the type of information circulating in the community while the stories in the library provide tacit social endorsement.

Given that low literacy is a key characteristic of the targeted user base, the E-Tool user interface is designed to be easy to use by a non-literate population. Drawing on the experience described in (Medhi et al. 2007), instead of text, navigation aids are provided using graphical icons, voice annotations, buttons with distinct colors and audio prompts to guide users through the various options. We stress that the UI is text-free, in line with the authors' findings that text may intimidate non-literate users. At the same time, many non-literate users have basic numeracy and readily recognise numbers. When using icons, the authors suggest that cartoon-like drawings are more readily understood than realistic pictures. A full context video dramatising how to use the application and what benefits can be obtained is also useful in explaining what the application does and in motivating the users (Medhi & Toyama, 2007).

The E-Tool user interface follows the design guidelines for non-literate users, with audio feedback prompts associated with all system actions and screen icons, the latter playing when the cursor "hovers" over an icon. Thus, the system is similar to an IVR in that it plays context specific prompts based on the event that has just occurred, guiding users through the interface by playing instructions about the available options at that point and the next action to take. If users hesitate, by moving the cursor over an icon they can hear information on that option. A help icon is also available on all pages. In designing the interaction and wording of the prompts, best practices from speech based applications are applied. While basic consideration was given to usability, this was not the focus of our research and thus received the minimal attention necessary to make the prototype operational. In choosing the icons, we chose images with which villagers were likely to be familiar such as radios, TVs and standard controls such as play, stop, etc.. Note that such standards must be localised for the target culture.

7.1.2 E-Tool functionality

The actual E-Tool consists of four screens corresponding to its main features:

1. The main page
2. The video feature (TV icon)
3. The story library (radio icon)
4. The interview feature (face icon)

Main page: This corresponds to the opening screen of the application, depicted in Figure 14. It offers the other three options to the user.

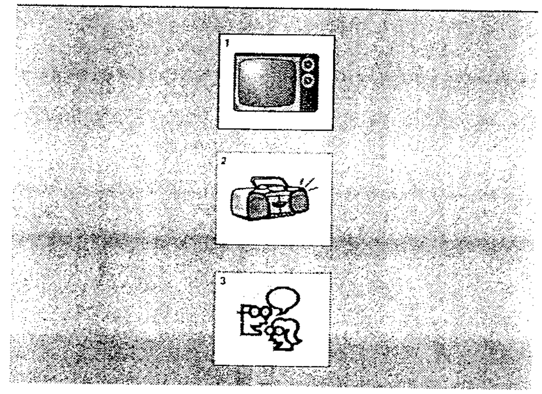

Figure 14. E-Tool main page

Video feature: This feature introduces the E-Tool and its current purpose. On selecting the TV icon on the main page, the system opens the video page depicted in Figure 15, and automatically starts playing the video. Users have the following options when watching the video:

- Pause the video (red button)
- Resume the video (arrow button)
- Scroll forward or backward to a new playback position in the video (scroll bar)
- Return to the main page (yellow button)

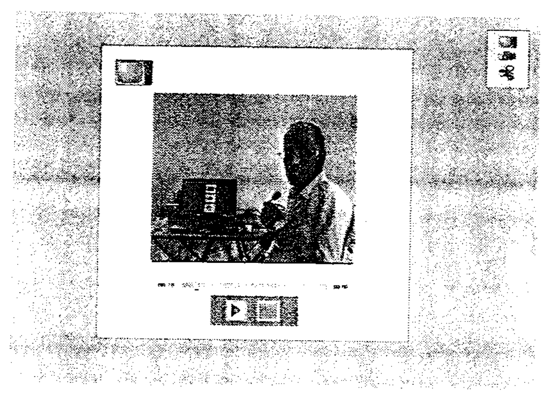

Figure 15. Video page

Story library: This feature lets users select and listen to a story from the collection of stories told by other members of the community. On selecting the radio icon on the main page, the story library page opens (depicted in Figure 16) and the user is advised of what he can do. Stories are classified as examples or regular stories, with examples presented at the top. Stories are identified by an automatically generated number when the story is created, and added to the end of the list. Optionally, a photograph can be associated with a story. This feature provides the following options::

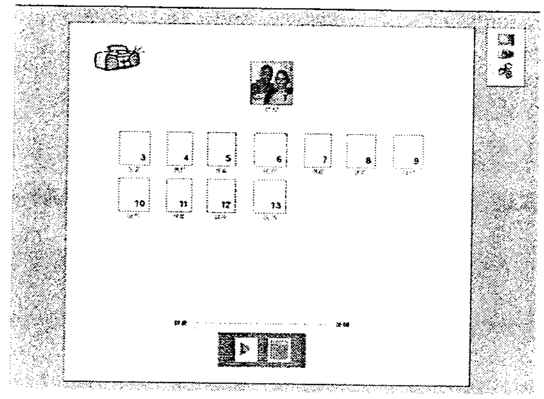

Figure 16. Story library page

- Hear the audio title of a story (move cursor over a story icon)
- Play a story (click story)
- Pause playback (red button)
- Resume playback (arrow button)
- Scroll forward or backward to a new playback position in a story (scroll bar)
- Return to the main page (yellow button)

Interview feature: This feature lets users record their own story. The feature guides users through a series of questions or themes without constraining the order in which they are answered. The feature presents a first question whose answer will serve as the title or identifier for that particular story. It then presents a set of primary themes, optionally followed by a set of secondary themes preceded by an introductory prompt. This is depicted in Figure 17, with the title at the top, the primary themes on the first row, and the secondary themes below. This structure supports eliciting answers from a second user, in the case where a story includes two perspectives. The tool supports a maximum of 13 questions, including the title question and both primary and secondary themes. When telling a story, users have the following options:

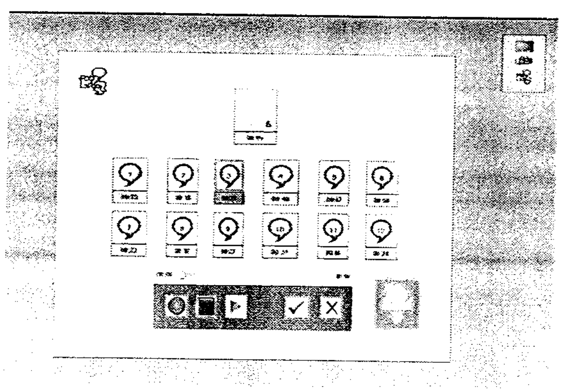

Figure 17. Interview screen

- Play (or replay) a question (hover cursor)
- Record an answer (green circle)
- Stop recording (red square)
- Play (or replay) an answer (arrow button)
- Record more at the end of an answer (green circle)
- Erase an answer (cross)
- Move on to the next question (check mark)
- Save a story to the story library (blue envelope at bottom)
- Return to the main page, with or without saving their story (yellow button)

On entering the interview page, the system automatically plays the first question then plays a short prompt advising the user to press the green circle when he is ready to speak, and the red square once he has finished. Visual feedback is provided to indicate the current question and the questions that have already been answered. When recording, a flashing "answer box" displays the length of the answer in minutes and seconds, while a scroll bar shows the current recording position with respect to the maximum length. Currently the maximum is set to a default of 5 minutes, after which time the intensity of the flashes increases. Users are given an additional minute to finish their answer before recording automatically stops. Once a user has finished recording an answer, to move on to the next question, he presses the check mark, to review the answer he presses the arrow button, or to erase his answer he presses the cross. When the user has answered the last question, he is advised how to save his story.

The user actions and system operations as well as paths through the four screens described above, are captured in the Use Case Map (UCM) presented in Figure 18. Following the semiformal notation proposed by Buhr (1998), UCM uses three basic elements to provide a visual representation of the various scenarios that comprise the interaction between a user and a system:

- **Rectangular boxes** represent run-time components such as software (objects, processes) and non-software entities (actors, hardware);
- **Lines** represent scenario paths, showing the progression of a scenario from start to end, respectively denoted by circles and perpendicular bars;
- **Crosses** represent responsibility-points along a path, indicating something to be performed (actions, activities, operations, tasks) by the entities.

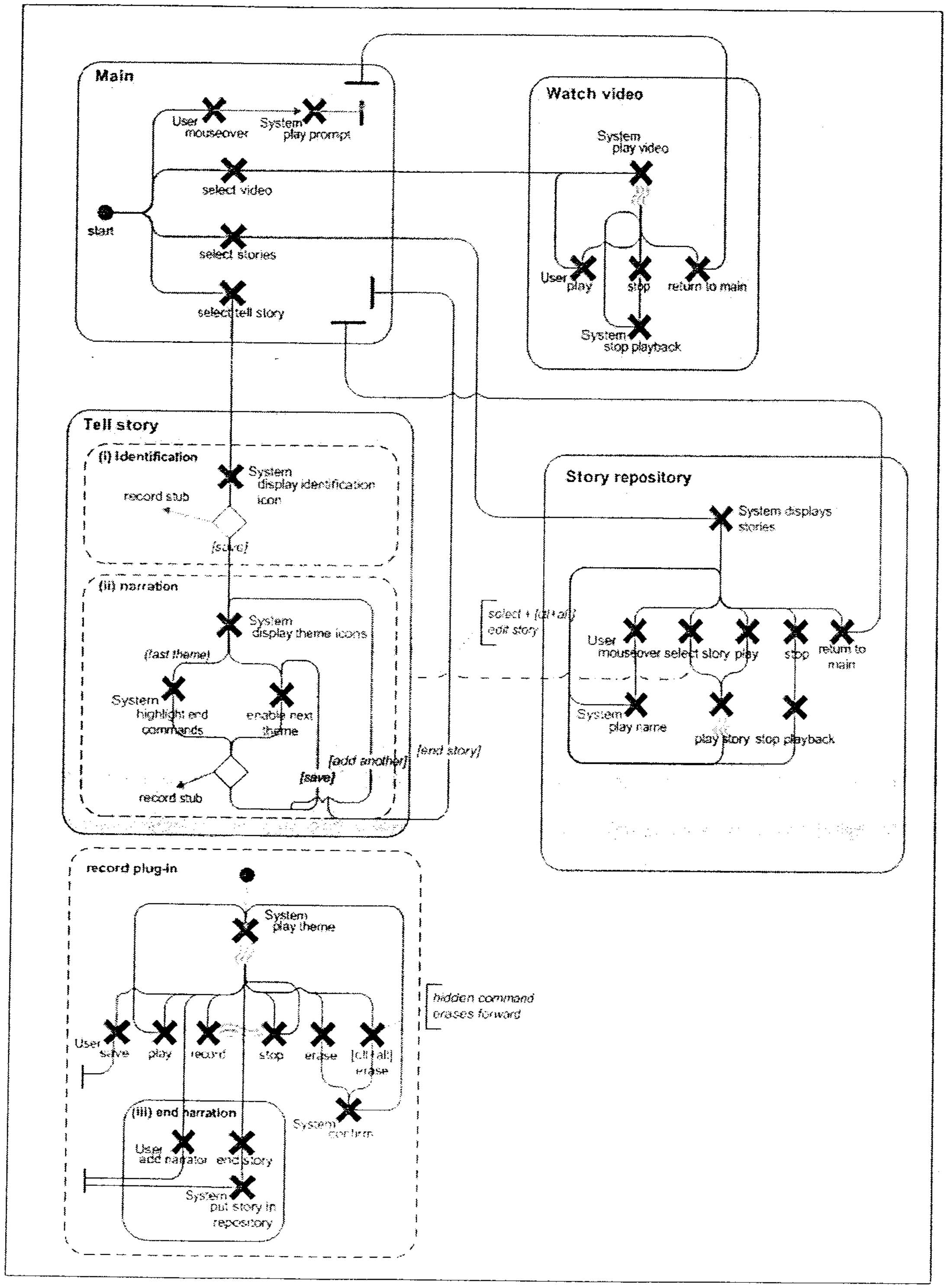

Figure 18. E-Tool Use Case Map

The E-Tool prototype was developed using the Microsoft Visual Basic 2008 Express Edition IDE, running on Windows XP Professional, with the MSDN DirectX and Windows Media Player 11 packages installed.

7.1.3 Administrative functionality

As this is a proof-of-concept, only the minimal administrative functionality necessary to facilitate the setup up and conducting of trials is provided. The following utilities assume the presence of a technician:

1. **Language selection:** On starting up the application, a prompt directory window appears, allowing the technician to specify the language that will be played. While the application is running, the technician can switch languages by pressing [CTL + L] in the application window (except for in the interview window). This latter ability is important for testing the application in a new language.

2. **Edit an existing story:** Sometimes, a user may not finish a story in one session, or would like to modify a story at a later time. To edit a story, the technician must first select the story in the library, and then press [CTL+E]. This will open the selected story in the interview page, where the user can proceed to edit his story using the interview options.

3. **Recording questions:** To record questions, minimal support is provided apart from the E-Tool itself. The technician opens the interview page, and records questions instead of answers. The technician must then copy the prompts to the appropriate directory for that language to be available.

In formulating the questions, while the application per se does not restrict the content or ordering of questions, the answer to the first question serves as the audio title for the narration in the story library. Consequently, we recommend that the first question be a directed, factual question that elicits a short response, such as the narrator's name or a title for the story. Making the first question short and easy to answer serves the additional purpose of inciting a spontaneous response, thus overcoming any initial anxieties users might have at the same time that they can familiarise themselves with the system's operation as well as check the quality of the recording and make any adjustments with regards to positioning the microphone.

7.1.4 Enhancements based on field tests

Below, we briefly describe some enhancements to the E-Tool based on our field experience. Enhancements pertaining to the tool's administrative functions are taken as a given and not included.

While a laptop computer is well suited for testing a proof-of-concept, it rapidly became evident that operating a computer, and more specifically the mouse device, required familiarity with the device. Although given enough time, villagers could become proficient with a mouse (as demonstrated by several users), since this was not our primary research objective, we simply put someone already proficient in charge of the mouse when recording. We recommend that in future versions the mouse device be replaced by an easier mode of interaction such as a touch sensitive screen or arrow buttons similar to those on mobile phones.

We also suggest the following improvements:

- Playback of a context specific help prompt after a certain amount of time has elapsed without any input could assist users who are hesitant with regards to what they should do next. While not strictly necessary if a technician is present, it would reduce the need to have such a person present.
- Currently the prototype does not address security concerns with regards to the confidentiality or privacy of the information collected. This would need to be addressed to support the collection of information of a sensitive or controversial nature.

7.2 Projected Future Version

While a laptop version of the E-Tool is a feasible solution, a laptop deployment presents certain inconveniences. Chief among these is the value and operation of the laptop itself, requiring the presence of someone trusted and trained in its use. Although this can be addressed by having a technician accompany the laptop, this does not fully satisfy the requirement of having a locally maintainable application that villagers can use themselves. Instead, we propose a mobile phone version of the E-Tool in which structured audio-stories are created using a handheld mobile phone (Pitula et al., in press). The high penetration and ubiquitous availability of mobile phones in both urban and rural areas of many developing countries make mobile phones an ideal platform for the application. In India, for example, mobile phones have a 65% market penetration rate[6] and even more people are familiar with their operation even though they themselves may not possess one. A mobile device is highly transportable and more readily affordable and accessible to a wide user base, while mouse related issues are entirely circumvented.

Using a mobile device, users can create stories as with the E-Tool. Similar to the desktop version, navigation is made simple by prompting and using familiar symbols. To facilitate human interactions, the phones can be equipped with a speaker-phone or ear-phone for listening to the speech output while viewing the screen. With enhancements, such as providing upload and download capabilities through the Internet, the SDS application can be remotely installed and configured and completed stories can be uploaded to a central server. In this way the collection process can be made more efficient and widely accessible to both participants and software analysts.

7.3 Potential Applications

The E-Tool, along with the related SDS elicitation technique and methodology, is specifically designed for eliciting needs in the context of rural ICT4D projects, and this is its intended purpose. However, there is no reason that the SDS approach cannot be applied to elicit needs in non-rural settings or for conventional software projects. The disadvantaged and marginalised are present throughout the world and literacy level do not necessarily reflect functional literacy. The E-Tool is ideal for such populations. The E-Tool can be applied to elicit needs from people such as the elderly, the disabled or the poor, who might feel uncomfortable or otherwise disinclined to participate in conventional studies. And even when dealing with literate users, as noted in (Gausepohl, 2008), there are certain domains such as healthcare, where access to end-users and the context-of-use is restricted. In such domains the E-Tool provides a means of collecting contextual information that otherwise would not be available. In other domains, such as consumer product development, the E-Tool can be used to obtain information directly from end-users, to complement the information collected through conventional techniques such as focus groups or ethnographic studies. Applying the E-Tool in this way would only require cosmetic changes, to localise its appearance for the targeted audiences.

---

[6] Telecom Regulatory Authority of India

More generally, the SDS elicitation technique can be used for collecting qualitative data in other research efforts (without necessarily applying the related SDS methodology to analyse the data). One obvious application is to collect evaluation data from end-users regarding their usage and satisfaction with an ICT4D system after it has been deployed. The E-Tool can also be applied in other fields that rely on qualitative data, such as sociology or psychology. Here, facilitators face comparable problems to those encountered by software analysts when conducting studies that involve participants from different cultures and literacy levels, and the E-Tool might prove of use. Social development interventions present another interesting opportunity where the E-Tool can be applied to elicit needs from the grass-roots with respect to establishing national and international policy. Currently, policy is often set in a top down fashion with input from experts because of custom and the difficulty of obtaining input from constituents and/or the targeted groups. In this context, the E-Tool can be used to collect information to produce a collaborative bottom-up analysis to complement the top-down perspective, as proposed by Kerr (2003). Here again the E-Tool can be applied as is, with minimal cosmetic changes.

The E-Tool also offers interesting possibilities as a tool for off-line communication and the sharing of stories. While the E-Tool design incorporates the goal of information sharing in a village setting, not being our primary research objective, this aspect of the prototype received less attention in our field tests. However, the E-Tool could easily be adapted to emphasise this aspect of providing a platform for sharing community information, events and concerns. With minimal changes the E-Tool could also support more conventional storytelling. For example, given some topic of interest (e.g. historical, cultural or other) the E-Tool could be used to collect and share stories on that topic. Another interesting area to investigate is getting input from end-users regarding content of interest in the context of telecentre projects. One of the shortcomings with the telecentre model is the difficulty of creating relevant content and keeping it up-to-date in an ongoing fashion. While NGOs and other organisations strive to provide content that is meaningful and useful to the disadvantaged people with whom they work, this is often a top-down process. Here, applying the E-Tool concepts, users could voice the information they seek, and potentially answer questions from other users. Finally, (and ambitiously), we can seek to better understand how the process of telling and listening to stories acts as a catalyst for change (Kerr, 2003), and adapt the E-Tool to better support this purpose.

Extending the E-Tool concept itself also offers interesting opportunities. In many countries, small entrepreneurs, such as vegetable sellers and auto rickshaw drivers, already use mobile phones in innovative ways to provide service to their customers. Drawing on the basic E-Tool capabilities (i.e. record prompts, collect audio responses, attach icons or images and play videos), with appropriate enhancements the E-Tool concept could be extended to support the creation of "mobile pages" similar to web pages, where entrepreneurs can e.g. advertise their services and collect orders from customers without reverting to text. Such text-free applications could be used for blogs, community pages, and other information services oriented towards non-literate, experiential populations.

Chapter 8: Case Studies — Field Work

In order to test the viability of the proposed SDS process for requirements elicitation, we conducted three field studies in two rural areas of India using the E-Tool described in the previous chapter. Our main objective was to determine if the SDS process can effectively be used for requirements elicitation within an ICT4D context by determining:

1. Is SDS usable by end-users?
2. Are the needs and constraints that emerge from the stories trivial or non-trivial; Are the findings obvious or non-obvious?
3. Is the process and use of the tool repeatable and adaptable?

For the sake of this field work, we assumed that our aim was to develop two software systems: (a) to support rural farmers in their farming; and (b) to support decisions regarding the education of children after Grade-12. Towards this end we elicited stories on two different topics, namely farming and the higher education of children, collected in two different rural regions with distinct regional languages in India. Below we first provide a brief description of the rural Indian context before describing our experiment and results. To show that this process reveals non-trivial information and demonstrate the application of the SDS process, we present an analysis of the farming stories, and demonstrate how the stories are transformed into an augmented goal-model suitable for elaborating software requirements pertaining to providing information on seed suppliers. We then define a measurement framework based on this goal model, to illustrate how the high-level goal-model can be used to define a set of success metrics for assessing a project's social impact as part of its evaluation criteria.

8.1 The rural Indian context

In India there are extreme disparities in socioeconomic conditions. Whereas the new economy centered on urban areas is giving rise to a comfortable middle class, it has only created aspirations for those from the social classes below. The traditional economy of the rural areas, largely based on farming, has left much of the rural population in extreme poverty with few options out (Robinson, 2007; Rezwan, 2009). Over 70% of the Indian population still lives in rural areas, many barely surviving on subsistence farming, seasonal work and occupations dictated by caste, with incomes below the international poverty line. Small plots, land depletion, poor yields, usurious lending practices and corrupt buying agents all conspire to put many farming households into a downward spiral of debt. A crop failure under such conditions has disastrous effects, pushing many farmers into despair, and in certain states suicide among farmers has reached crisis proportions. Although the government tries to regulate many aspects of farming, unregulated and corrupt practices persist. Villagers view higher education as the only way for their children to aspire to a better future. However, while primary and secondary education is government funded and locally accessible, higher education at the University or Technology level is not, and some private institutions are of questionable merit. Word-of-mouth is the only source of information for many parents and children regarding career choices and available education programs. This limitation is driven by both the cost of accessing up-to-date and reliable information sources as well as the people's culture. We focus our study on the following two problem areas, namely:

1. The problems faced by farmers, and
2. The problems faced by parents and children when considering higher education.

8.2 The experiment

A total of three studies were conducted—two near Chennai in Tamil Nadu state and one near Bangalore in Karnataka state. All three took place in farming villages typical of the rural Indian context. In Tami Nadu, stories on farming and higher education were elicited while in Karnataka the topic was farming only. Altogether 30 stories were collected, 17 on farming and 13 on higher education. These were told by both male and female participants representing a broad age range, from children to the elderly, and a cross section of financial conditions, from the very poor to those considered well off by local, rural standards. In Tamil Nadu, 12 farming stories were collected on the main street of a village and in local homes over a two-day period. In Karnataka, 5 stories were collected in the office of a local NGO involved with farmers over the course of an afternoon. The stories pertaining to education were collected in Tamil Nadu at two local schools over two days.

The farmers who participated were primarily male, with only 3 female participants. In Tamil Nadu they were recruited informally off the street, whereas in Karnataka the NGO invited farmers with whom they regularly work. Participants were 30 years old and over, including two over 60. While one participant had never attended school, the majority had completed some schooling, but only 4 had a higher education (mainly agronomy) and spoke broken English. The size of the farmers' plots varied from 1 to 10 acres, and farming was the primary activity of all but two. Their financial situation varied from very poor to comparatively well-off. Only two did not need to borrow money to finance their farming activities from crop to crop. In a typical year, farmers might have one, two, or a maximum of three crops, each taking a variable number of months to grow.

In the case of higher education, both the parent and child participated in telling a story. The children consisted of 5 girls and 8 boys aged 15–17, all in grades 11 and 12 of high- school, and almost all were among the top students in their class. This, along with the availability of a parent to participate together with the child, constituted the recruitment basis for our study. A higher number of mothers participated, likely due to the sessions taking place at local schools in the middle of the day. None of the parents had a higher education themselves, all were employed in typical, traditional activities earning daily wages, and some were from among the poorest in the village.

In all three studies, local people assisted in setting up the study, recruiting participants, and identifying relevant themes, and local leaders were approached to get their tacit approval and support. In the case of farming, the focus was on collecting general information (family, land and water; crops cultivated; seed; use of fertiliser and pesticides; manual and mechanised labour; and financial aspects). Regarding education, parents were asked to provide background information about themselves, their child, the subject they wanted their child to pursue, financing higher studies and what information they might need. Children were asked about what they liked doing, their school, what career they envisaged and what studies this would require. The themes were formulated into open-ended questions, translated and recorded in the local language using the laptop computer. Respected individuals from the community, or alternatively representative users, were then asked to record their story to serve as an example.

8.2.1 Protocol

The actual interviews were conducted using the E-Tool running on a laptop computer, with an attached mouse and hand-held microphone. All E-Tool prompts were in the local language. We observed that while the participants were all adept in using mobile phones and some could even "text", they had considerable difficulty moving and using the mouse. Consequently, a technician familiar with the E-Tool took charge of the mouse and selecting the appropriate controls. As the technician did not speak the local language, any supplementary explanations and instructions were provided in English and translated by

Figure 19. Collecting stories on the main street of Villivalam in Tamil Nadu state.

either the facilitator or others present. However, as what was expected was readily understood from hearing the sample stories, watching the video and observing others using the E-Tool, the facilitator's interventions were limited to positioning the microphone and hand-waving to indicate start and stop speaking. All the participants told their stories in their regional language.

The actual sessions were conducted informally and group participation was encouraged. On arriving, participants were either shown the video or given a brief explanation of the purpose of the study in the local language. They were then shown how to operate the E-Tool and invited to record their own story. Each participant recorded his or her story in turn while the others listened. Late arrivals quickly picked up what was going on by observing the others. On completing their story, participants were photographed and offered a small gift (monetary or a box of candy). This photograph would serve to identify their story within the story library.

8.3 Outcome of experiment

The acceptability of the SDS approach, assessed in terms of ease of use, ease of learning the tool and the natural flow with which participants told their stories, exceeded our expectations. All the participants were able to tell their story and were enthusiastic about doing so. Villagers participated readily and quickly picked up the operation of the E-Tool. Once they began talking, they became engaged in telling their story and were not distracted by the mechanics of recording. While in almost all cases they participated in groups, their stories were highly personal and did not show any signs of "groupthink". At the same time, the group provided an audience for the teller, making the narration a natural communicative exchange. Interestingly, when we first described our study to the personnel of the Karnataka NGO, they were convinced that an informal, group approach could not provide the information we sought. They advised us, based on their experience, that we must interview each farmer in depth individually, as otherwise "*you won't get the answers you want*". Nonetheless we proceeded with our experiment, following which the NGO personnel expressed their astonishment at the richness of the stories collected, contrary to their expectations.

8.3.1 Analysis of farming stories

Our analysis of the stories indicates that they are highly useful in identifying the participants' concerns and reveal an abundance of contextual information. While a full discussion is out of scope, to support this position and demonstrate how the SDS process is applied, we present examples from our analysis of the farming stories, showing how we derive a set of non-functional and functional requirements pertaining to providing information on seed suppliers. The transcriptions of the famers' stories and the detailed analysis are provided in (Pitula, 2009). As illustrated in Figure 20, the information extracted from the transcribed stories is fully sufficient to construct a meaningful domain model highlighting the major concepts, concerns and relations.

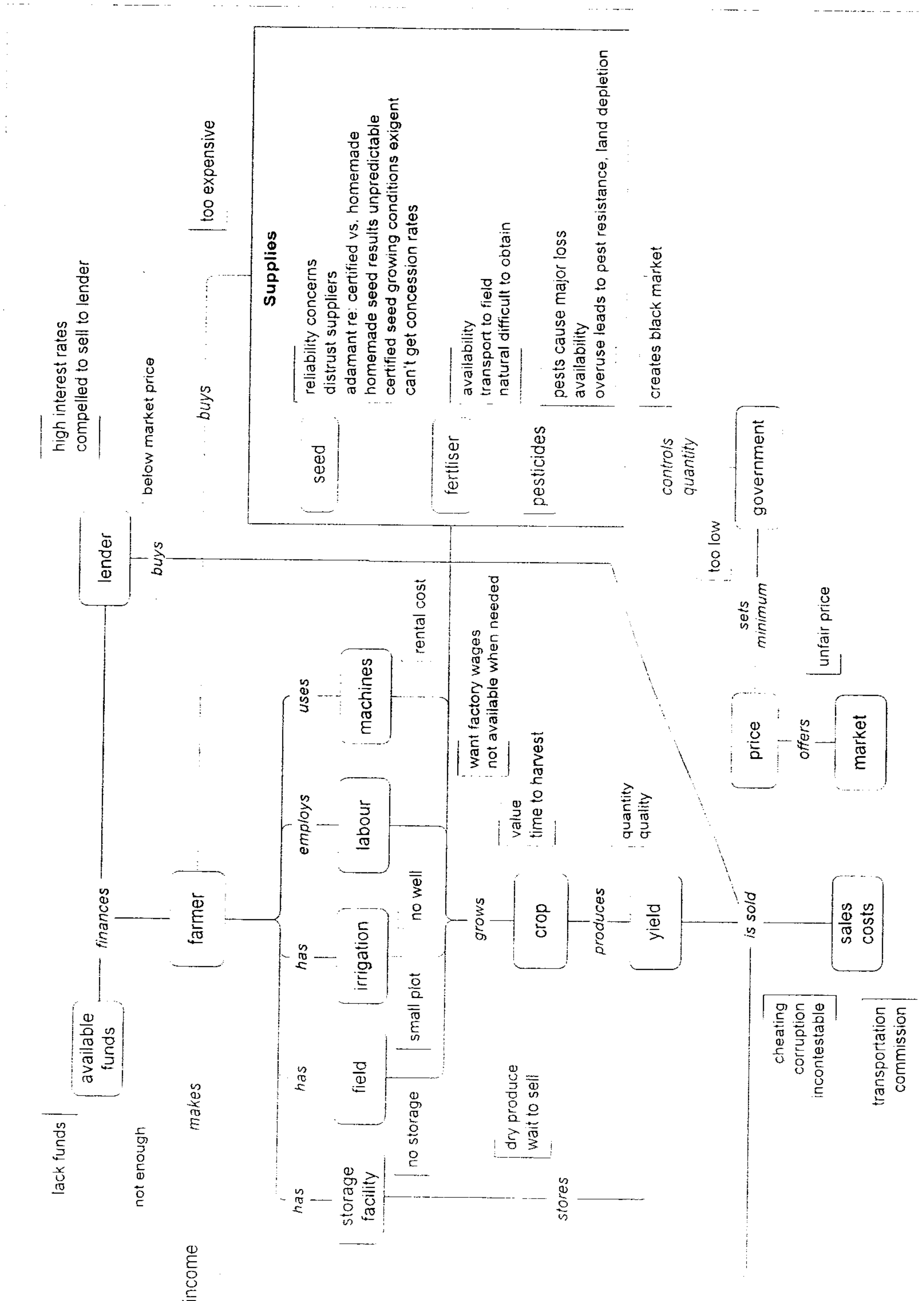

Figure 20. Problem domain model describing farming

Based on the cultural analysis indicating strong experiential tendencies, we established the related cultural inhibitors (i.e. averse to disruptive change; reluctant to experiment; rely on knowledge conveyed through concrete experience; not predisposed to speculation or abstract thinking; reluctant to act individually; and favour high-context personal communication).

From the prioritised list of issues, profitability emerges as the farmers' primary concern. Modelling profitability as a decision situation with profit as the variable to optimise produces the influence diagram depicted in Figure 21. Briefly interpreted, profit is income less expenses, with expenses functionally determined by decisions concerning supply purchases, which in turn influence the yield (i.e. what seed, fertiliser and pesticides are applied influence the quantity and quality obtained). Supply decisions are influenced by decisions regarding crop (i.e. different crops have different requirements), and both crop and supply decisions are influenced by financing decisions, which in turn are based on available funds and/or loans. Examining the income path, income is functionally determined by the sales decision, informed by the market price (itself informed by the government set price), the ability to store produce (thus dry the produce and wait for better prices), and the yield, with the sales decision influenced by the financing decision (given that farmers who have taken out loans are obliged to sell to the lender).

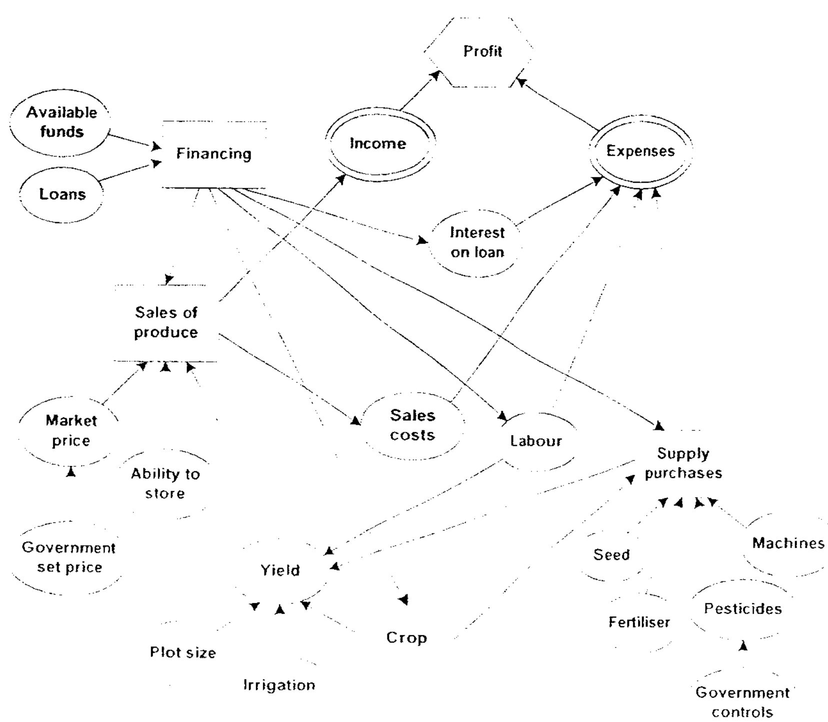

Figure 21. Influence Diagram displaying the causal relationships among the factors contributing to a farmer's profitability.

A cursory analysis of the influence diagram reveals significant external constraints on the farmers' choices exacerbated by serious negative feedback loops. The profile analysis related to profitability establishes a difference between small versus large landholders (less than 5 and 5–10 acres respectively). Small landholders typically lack funds, have small plots, no wells, and no storage facilities. Mapping the consequences of these external inhibitors produces the causal loop diagram depicted in Figure 22, with four major feedback loops described below.

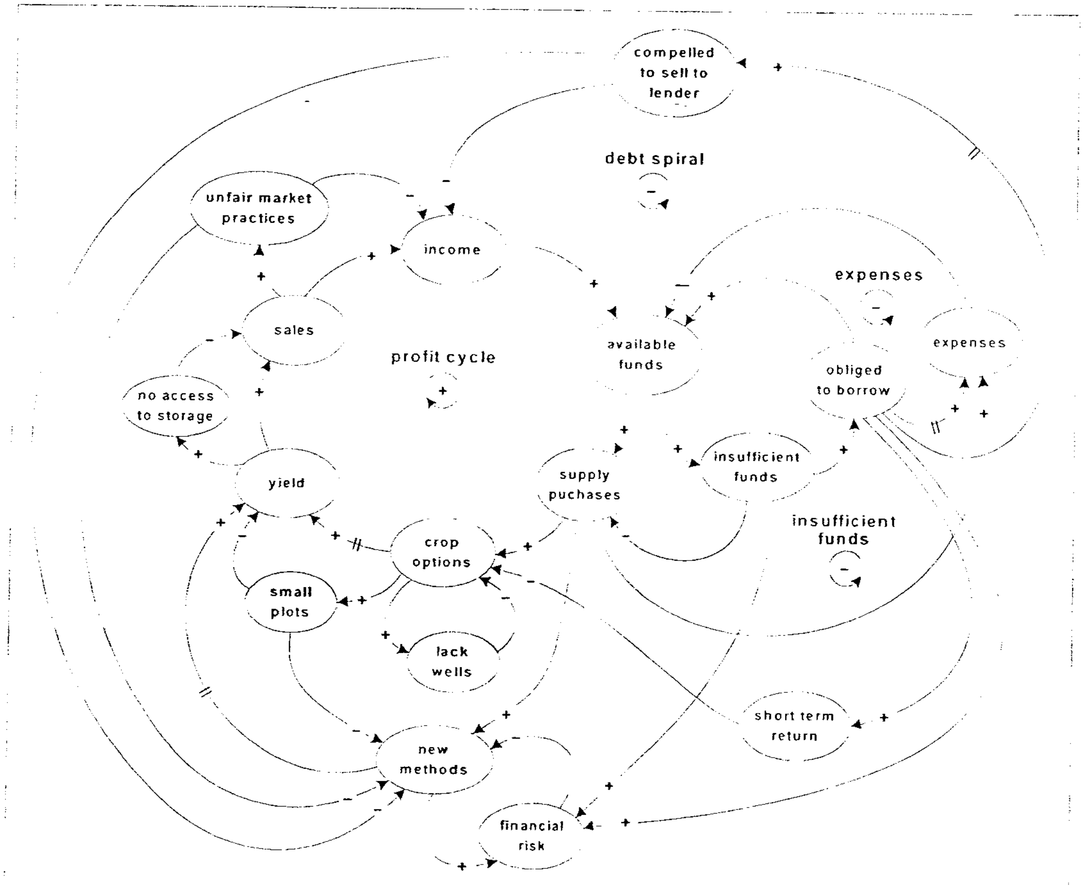

Figure 22. High-level causal loop diagram of the farming context

The positive *profit cycle* consists of farmers spending available funds to purchase supplies to grow crops, producing a yield whose sale provides an income, increasing available funds. Concurrently, the purchase of supplies incurs expenses that are deducted from available funds, resulting in a negative *expenses loop* that balances the profit cycle and maintains equilibrium. However, while there is a delay between growing a crop and obtaining a yield, the effect of expenses on available funds is immediate. Effectively, farmers must invest in their crop upfront and only obtain an income after the yield is sold. Farmers with insufficient funds will reduce their supply purchases (cheaper seed, less fertiliser, less pesticide, etc.) to the detriment of their crop, and have a higher financial risk, thus discouraging them from experimenting with new methods, to the overall detriment of income. Over time, this *insufficient funds* loop has a negative reinforcing effect on available funds through lower yields, land depletion, etc. resulting from inadequate investment in the land. While certain farmers may persevere in this declining state, others are obliged to borrow, putting them into a negative *debt spiral* from which few recover. Borrowing money incurs high interest charges, increases financial risk, impels farmers to seek short term returns (to pay off the loans), and compels them to sell to the lender at prices below the market rate, all of which have a deleterious effect on income. Lacking funds to begin with, the reduced income they make is insufficient, obliging them to borrow in an ever downward spiral of debt. All of these constitute hard constraints that influence the farmers' choices and impede their ability to act.

8.3.2 Constructing a solution for a specific issue

In order to show the utility of our models, we now demonstrate how they can assist in selecting an appropriate ICT4D intervention and identifying the relevant inhibitors and enablers (or non-functional requirements), based on which a set of high-level functional requirements is derived. Towards this end, we first construct a high-level goal model that lays out a set of potential interventions for attaining some desired goal. Consulting the influence diagram and causal loop diagram, we identify subgoals whereby the desired social impact can be achieved, given the characteristics of the targeted farmers and the inhibitors affecting them. Using a detailed goal model, we then refine the chosen subgoal with respect to the inhibitors, to arrive at a set of high-level functional requirements related to that subgoal. For the purposes of our example, we pursue the relatively unconstrained subgoal of providing information on seed suppliers.

Having established that profitability is the farmers' primary concern, we set the goal of improving profitability. From the related influence diagram (Figure 21), we derive the high-level goal-model depicted in Figure 23. This indicates that profit can be improved by increasing income or reducing expenses, with an increase in income achievable by improving sales options (better prices, less cheating, etc.), yield (new methods, higher quality produce, etc.), and crop type (better varieties, more valuable crops). However, sales are influenced by financing choices, and lacking funds, many farmers are obliged to borrow and thus they are compelled to sell to the lender. Consequently many will be unable to benefit from improved sales conditions. Attempts to improve the yield or crop will encounter the same barrier, especially if these involve additional costs. Effectively, farmers who are in—or susceptible to falling into—a debt spiral lack funds and cannot assume financial risk. This deters them from improving their yield and impels them to seek short term returns, limiting their choice of crop. Thus, any solution that seeks to increase income, must counter the obstacles emanating from insufficient funds and the debt spiral.

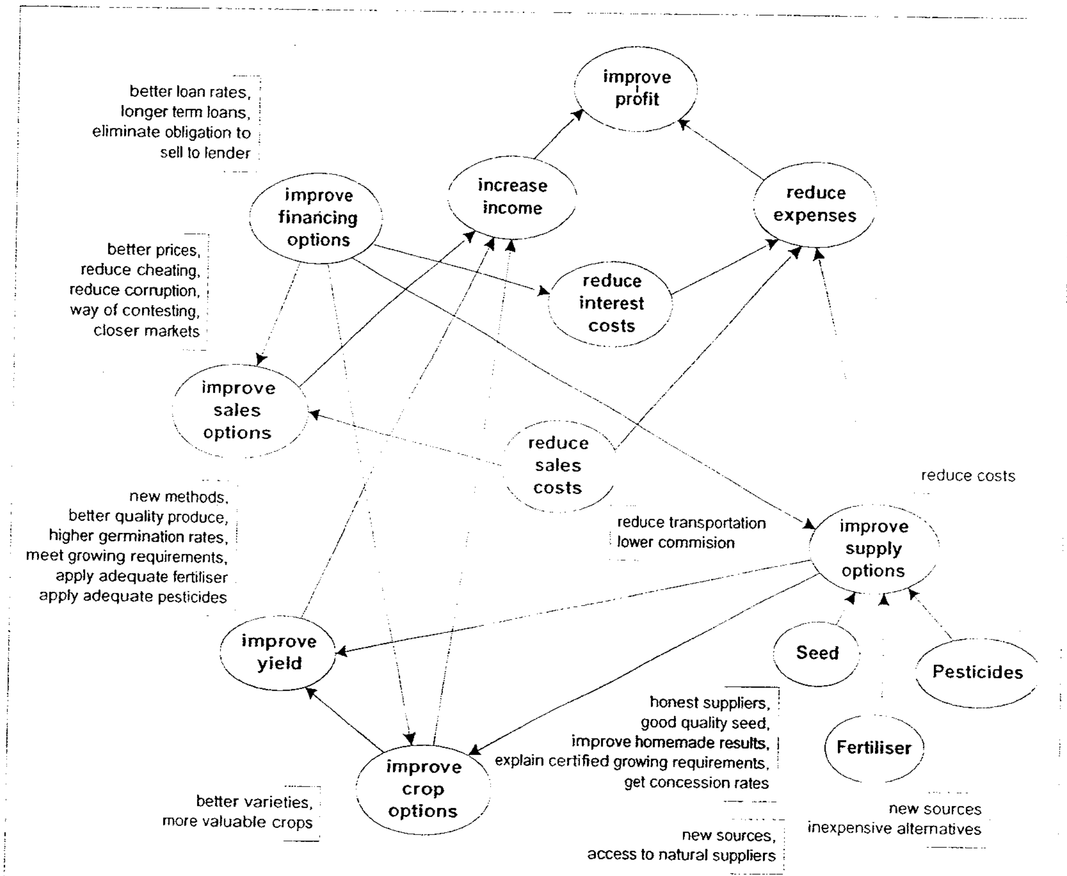

Figure 23. High-level goal-model derived from the influence diagram presented in Figure 21, annotated with potential ways of addressing the concerns mentioned in the stories

Analysing the goal-model in this way, potential interventions are assessed and viable subgoals elaborated. For example, providing information on "honest" seed suppliers (assuming "honest" suppliers exist) can reduce supply costs (through better rates) and potentially increase yields (with better quality seed and germination rates) without introducing additional expenses. The prioritised issues concerning seed indicate that reliability is the major concern, with many farmers distrusting suppliers because certain suppliers are dishonest. Expanding the supplier branch, distrust is likely reinforced by the cultural inhibitor of low-context communication between suppliers and farmers. The farmers' inability to assume financial risk acts as a further inhibitor. To offset these inhibitors we introduce the enablers "no financial risk" and "trusted suppliers".

Both are reinforced by "positive concrete experience", which itself is positively reinforced by "observable results". Trust in suppliers can also be reinforced by "collective participation" (i.e. I myself have no opinion about the supplier, but others that I know have trust), or by "high-context personal communication" with the supplier (i.e. he or she shares my context and is someone that I can relate to within my operational frame of reference). Thus, the goal of providing information on seed suppliers can be achieved by achieving the subgoal of "trusted suppliers", reinforced by "collective participation", "high-context communication" and "concrete experience", with the latter reinforced by "observable results". This is depicted in the detailed goal-model presented in Figure 24.

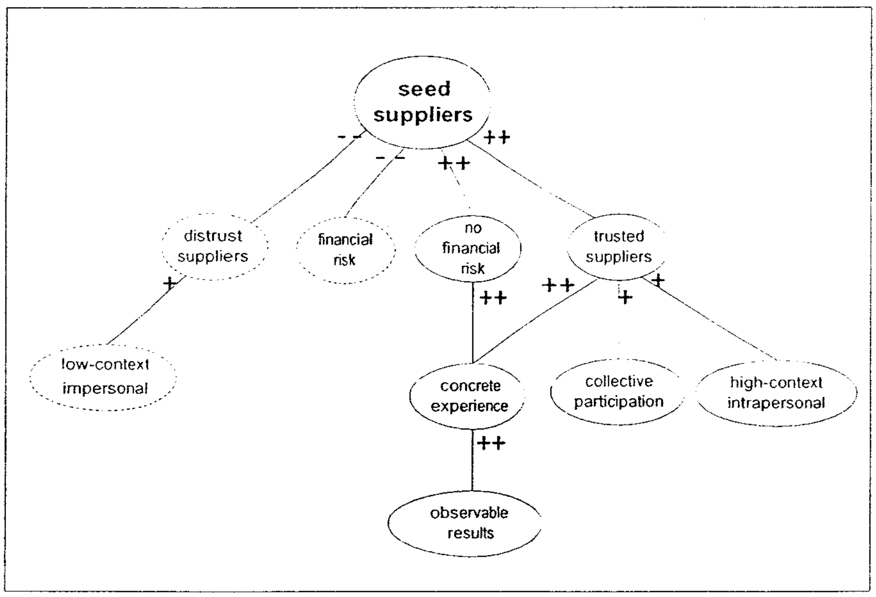

Figure 24. Detailed goal-model pertaining to providing information on seed suppliers with inhibitors in dashed ovals and enablers in solid ovals.

Once the enablers are laid out, different ways for realising them are considered. For example, observable results can be obtained by cultivating a demonstration plot in the village or by handing out seed samples to individual farmers to grow. However, such approaches are not readily scalable, and in the case of samples, the farmers' reluctance to experiment and act individually may limit their effectiveness. Videos showing demonstration plots might be a possible solution. We can also look at ways of increasing collective participation. One possibility is to establish communication among farmers who plant a particular seed variety, effectively creating a "user group". Another is to provide testimonials from satisfied farmers, presented in a situated, high-context, intrapersonal way, thus providing a form of indirect experience. The more recognisable the context and the farmers appearing in them, the more trustworthy, with personally known people the most trusted of all. Such an approach, which draws on the villagers themselves to appear in the content, has proven to be successful for building trust in agricultural extension work, with the possibility of appearing acting as a strong incentive to participate (Ghandi et al., 2009). Trust can be further reinforced by high-context, intrapersonal communication with suppliers, through e.g. village meetings, telephone or video conferencing. Thus, each enabler can be associated with multiple alternative sets of features, giving rise to the detailed goal-model illustrated in Figure 25, with the set of possible features (or functional requirements) derived from the constraints extracted from the stories depicted in rectangles at the leaves. These constitute additional requirements to the primary goal of providing information on seed suppliers.

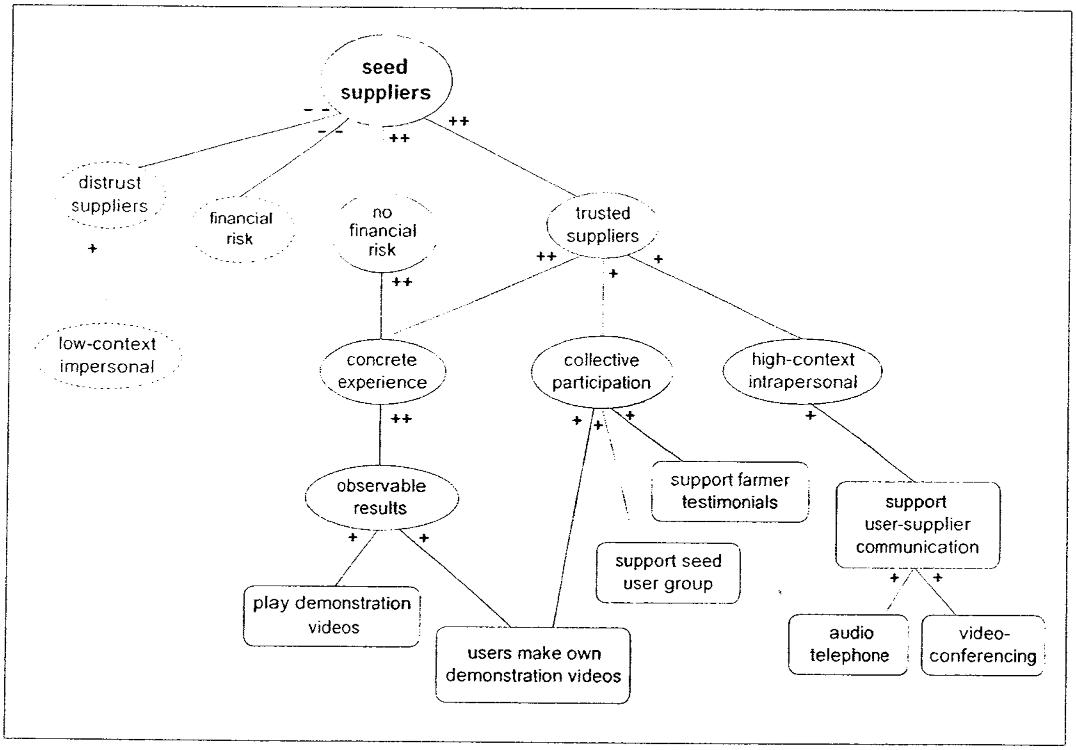

Figure 25. Elaborated goal-model pertaining to providing information on honest seed suppliers

In elaborating the requirements further, we consider how the various properties from our ICT4D quality model might be satisficed in the project design. In satisficing these properties, a number of new design constraints and requirements emerge. We briefly describe these below, providing only a concise elaboration as a comprehensive treatment is beyond the scope of this thesis.

- **Feasible**: Because of their lower exigencies with regards to infrastructure, skills and resources, the information service pertaining to seed suppliers should be delivered on a mobile device. While the service might be offered on a PC, particularly in areas with active telecentres, the higher exigencies with regards to acquiring, operating and maintaining PCs in rural villages would impede the service's rate of diffusion. On the service delivery side, the service could be maintained and managed by national, regional and local agricultural extension programmes or NGOs.

- **Affordable:** As with any new service, users should only be charged once a service has demonstrated its value. Initial funding options include agricultural extension programmes and external aid. Among the business models to investigate, the service might be offered under an online advertisement model, with seed suppliers advertising their offerings at no charge to users (with pricing according to the size of the business to ensure equitable access by both small and large suppliers). With regards to accessing the mobile device itself, while certain farmers may possess a mobile phone, others may not. These could be served by small entrepreneurs already providing mobile telephony services, or shared devices might be made available through local community initiatives such as farmers' cooperatives or NGOs. In all cases, pricing should be adjusted to ensure affordability by the poorest.

- **Accessible:** A mobile device with a text-free interface in the local language would make the service accessible to farmers from all socioeconomic segments and ethnicities. A comparable effort is required to ensure accessibility by seed suppliers of all sizes, including small local efforts, which, through advertising, could grow. A suitable process and interface is required to allow suppliers to be posted on the service.

- **Relevant:** The topic relates to a critical aspect of the villagers' primary economic activity, namely acquiring inexpensive, reliable seed for farming.

- **Trustworthy:** To assure the accuracy of claims made with regards to the crops produced with seed from different suppliers (i.e. seed reliability), along with positive testimonials, farmers must be able to register complaints and negative reviews so that suppliers of unreliable seed can be weeded out. Farmers who lodge such complaints must also be assured that any suppliers thus excluded cannot retaliate.

- **Beneficial:** Ideally, the farmers will have access to less expensive and more reliable seed as well as more seed varieties and potentially, different crops.

- **Sustainable:** With an appropriate business model, the proposed project can evolve into a cost-effective service. Acquiring seed is an activity that farmers repeat on an on-going basis. With suitable support, the service can be expanded to address emergent needs with regards to seed (e.g. placing orders, making reliable seed oneself, etc.). The use of mobile devices makes this project readily scalable, while enhancements such as download and upload capabilities through the Internet, would minimise local maintenance requirements.

- **Supports a community of practice**: the demonstration videos, farmer testimonials and seed user group constitute a learning 'space and place' where users can first observe before trying the service themselves. An introductory video dramatising how to use the service and the possible benefits can also be provided. The seed user group and the ability to give testimonials and create demonstration videos promote participation in a community of practice. These will initially be produced by designated "masters", but local farmers will be encouraged to contribute, thus expanding the community.

- **Culturally appropriate**: the service fits the traits of an experiential culture and is culturally appropriate as these aspects are incorporated into its design. The narrative and visual information provided by the service is suitable for the type of information circulating in the community while the videos and testimonials provide tacit social endorsement.

8.3.3 Conclusions drawn from the experiment

With this example we have shown that the SDS process provides non-trivial and non-obvious contextual information that is useful in deriving software requirements. Did the SDS methodology provide additional insight to the conventional RE process? To answer this question a reference point is required. To serve as such, we use an example arbitrarily drawn from the many seed supplier websites available on the internet and intended for users from very different sociocultural backgrounds. An excerpt from a government sponsored website is shown in Figure 26. A cursory examination of the site reveals that among the notable differences, along with assuming highly literate and internet savvy users, there is an explicit disclaimer regarding the endorsement or recommendation of any supplier, nor is consideration given to addressing concerns about financial risk. Information regarding specific suppliers is impersonal, limited to a contact address, website or phone number. There is no support for collective participation by users or provision of observable results. Instead, using the search function, users can access an extensive set of "scientific" documents containing the relevant search terms. While this site is not necessarily the "best", it is exemplary of the types of sites available, and clearly intended for literate users from an analytic culture.

The features derived from our analysis of the stories are in direct contrast to those present, or in this case absent, from the government website. From the stories, "trust" and "financial risk"

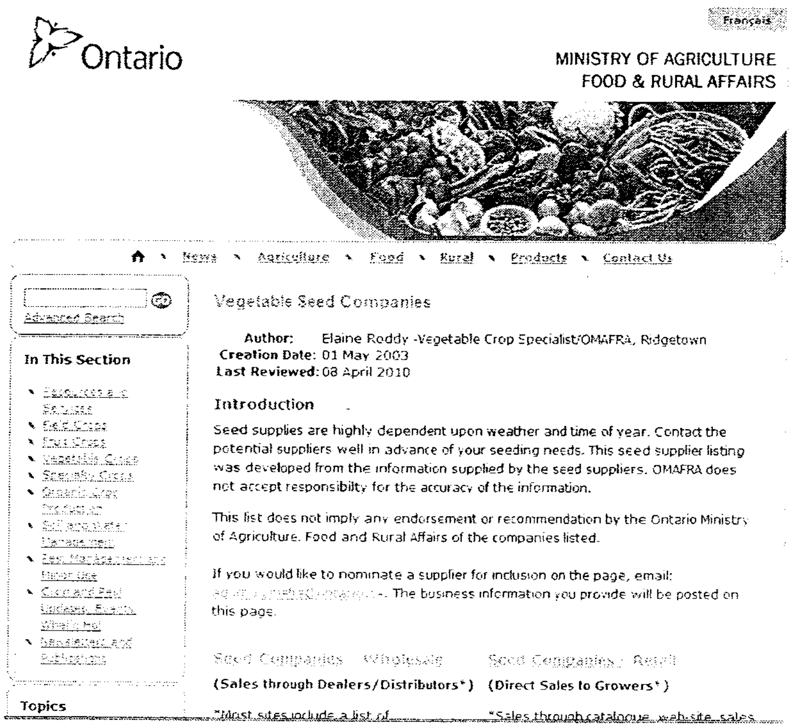

Figure 26. Website pertaining to seed suppliers for a different sociocultural audience

emerge as two critical properties to address for information on seed suppliers to be acceptable to its intended users. Given that users are from an experiential culture, drawing on our cultural model, we identify the enablers "collective participation", "high-context interpersonal communication" and "concrete experience", with the latter obtained through "observable results", as the means of building trust and reducing perceived financial risk within an experiential society. Operationalising these, we arrive at the functional requirements of supporting personal communication between users and suppliers, supporting seed user groups and farmer testimonials, and playing and creating demonstration videos—features that are notably absent from the government website. While the correctness and importance of these features cannot be accurately assessed without a deployed system, they appear eminently appropriate, based on the shortcomings and strengths documented in the ICT4D project literature.

This study clearly demonstrates that storytelling is an effective and acceptable way for people to express their concerns and needs with regards to some problem regardless of their literacy level. While in principle such a study would have been undertaken jointly with a development organisation, for logistic reasons this was not the case. Consequently, development experts were not consulted when selecting themes or analysing stories, nor was the analysis conducted with respect to any well-defined development goals. Instead, for our example we selected a subgoal unconstrained by external inhibitors requiring interventions at the policy level (e.g. changes to lending practices, the market structure or access to funds). Nevertheless and despite our limited knowledge of local languages, customs and the farming domain, we were able to develop an informed understanding of the problem that goes well beyond what could be discovered during an equivalent period of "scenic fieldwork". Additionally, by conducting this experiment on different topics in different regions and languages, we demonstrate that this approach is readily adaptable to new contexts in terms of population group, language, and topic.

We also mention some practical findings regarding applying SDS in rural areas. Many of the narrations were high-context, omitting contextual information that participants share in common and assume known. As the narrator's effusiveness appears to be correlated to his or her socioeconomic situation, we recommend recruiting participants from different socioeconomic strata to ensure sufficient background information is collected and to provide broad coverage for comparative purposes. Also, participants do not restrict their answers to a theme when it is brought up; instead information is spread across the entire narrative. Therefore the analysis cannot be partitioned by theme as information relevant to a particular concept may be spread across the entire story.

8.4 Summary of steps applied in the SDS process

Our example highlights the findings from our analysis and the models produced without going into the specific steps applied to abstract, interpret and model the information. Below we briefly recap the steps in the SDS process by which we arrived at these findings. Although the steps are presented in sequence, we emphasise that this is a highly iterative process of discovery.

- After the stories were collected, they were first translated and transcribed before being analysed to abstract the concepts, concerns and relations. A preliminary analysis established the domain concepts, the participants' profiles and the incidence of problems and conditions. A prioritised list of issues was produced with profitability emerging as the primary concern.
- The abstracted information was used to construct an augmented domain model of the problem domain (Figure 20), retaining links to the specific stories in which the elements are mentioned.
- The dominant cultural tendencies and associated cultural inhibitors were established by applying the cultural model to analyse the stories. The corresponding enablers comprise non-functional requirements related to the users' culture.

- The primary concern of profitability was analysed in depth to identify the set of contributing factors. These were established by examining the relations between the issues and the characteristics and conditions of affected farmers.

- Using an influence diagram, profitability and the set of contributing factors were modelled as a decision situation to show the flow among the elements and the influence of the various variables, decisions and outcomes on the primary issue (Figure 21). The resulting model describes the static structure of the problem.

- Using a causal loop diagram, we then modelled the various paths through the influence diagram and identified problematic feedback loops within it (Figure 22). With this model of the problem's dynamic behaviour we identified the external inhibitors and enablers affecting farmers. These correspond to additional non-functional requirements arising from the users' local context.

- Based on the influence diagram and the concerns depicted on the domain model, we constructed a high-level goal-model describing possible improvements corresponding to potential ICT4D interventions (Figure 23).

- Analysing the high-level goal model with respect to the external inhibitors and enablers , we identified improved seed supply options as a potential intervention. We expanded this subgoal by incorporating enablers (or non-functional requirements) to address the external and cultural inhibitors previously identified, producing the detailed goal-model depicted in Figure 24.

- We then considered different ways of operationalising the subgoals associated with the enablers, producing the elaborated goal-model depicted in Figure 25 with potential system features or functional requirements at the leaves.

- Finally, we applied the properties from the ICT4D quality model, to assess the solution with respect to local conditions in terms of the barriers to its use and sustained acceptance. From this analysis, a number of potential functional requirements emerged. For example, the property "accessibility" lead to the high-level functional requirement of allowing suppliers to be added to the service, while the property of "trustworthiness" lead to allowing farmers to register complaints and negative reviews.

8.5 Measurement framework

We now illustrate how the goal-models produced by our process can be used to define a set of metrics for assessing a project's success based on its social development goals. In defining these metrics we apply the 5-level Goal-Question-Metric evaluation framework proposed by Potts Steves and Scholtz (2005). For our example, we consider a system for delivering information on seed suppliers with the high-level goal of improving farmers' profitability, as described in our analysis. We identify three main areas of interest with regards to assessing the changes that the proposed system is expected to bring, relating to: (1) seed supply options; (2) farmers' profitability; and (3) what seed suppliers and seed the farmers actually choose. Note that depending on the team's interests, other areas might be selected, such as the impact on what crops are grown or farming practices. Each of these areas is restated as a system goal, with associated evaluation objectives, conceptual metrics and measures as well as implementation specific measures (IMS). In elaborating the measurements we draw on the high-level goal-model depicted in Figure 23 and the elaborated goal model of Figure 25, including all the features depicted in the leaves. The resulting measurement template is presented in

Table 5. We note that the project's technical success is a necessary precondition for assessing a project's overall success (i.e. the software must work and deliver information for it to be able to contribute to any significant social impact).

Table 5. Measurement template produced using the 5-level GQM evaluation framework

Goal statement 1	**Delivering information on seed suppliers will improve the farmers' options regarding what seed they can buy.**
Evaluation objective 1.1	*Assess changes in available seed supplies*
C-Metric 1.1.1	Did the set of seed suppliers available to the farmers change?
C-Measures	a) Changes in seed suppliers prior to and after intervention (suppliers available, actually used) ISM: new suppliers made available through the ICT tool ISM: new suppliers used by farmers
C-Metric 1.1.2	Did the available choice of seed change?
C-Measures	a) Choice of seed (crop, variety) prior to and after intervention ISM: new crops, varieties discovered through the ICT tool ISM: new crops, varieties tried by farmers
C-Metric 1.1.3	Did the seed prices available to farmers change?
C-Measures	a) National and regional high, low and average price asked for seed (type,

	variety) b) High, low & average price of seed available to farmers prior to and after intervention ISM: high, low and average price asked for seed by suppliers posted on the ICT tool
C-Metric 1.1.4	Did the overall price suppliers charge for seed change?
C-Measures	a) National & regional high, low & average price paid for seed (type, variety) b) Difference in price proposed to farmers prior to and after intervention with respect to regional and national prices.
Evaluation objective 1.2	*Assess changes with respect to reliability of seed supplies*
C-Metric 1.2.1	Were the suppliers "honest"?
C-Measures	a) Farmers' rating of suppliers' honesty ISM: farmers' rating of honesty of suppliers posted on the ICT tool b) Suppliers' pricing c) Regional and national seed prices (C-Metric 1.1.3) d) Yield obtained (C-Metric 2.2.1) e) Quality of crop (C-Metric 2.2.2)
C-Metric 1.2.2	Was the seed of good quality?
C-Measures	a) Farmers' rating of seed quality ISM: farmers' rating of seed quality obtained from suppliers posted on the ICT tool b) Yield obtained (C-Metric 2.2.1) c) Quality of crop (C-Metric 2.2.2)
Goal statement 2	**Delivering information on seed suppliers will improve the farmers' profitability.**
Evaluation objective 2.1	*Assess changes in seed supply costs (expenses)*
C-Metric 2.2.1	Did the amount farmers pay for seed change?
C-Measures	a) Changes in amount paid for seed ISM: amount paid for seed from suppliers posted on the ICT tool
C-Metric 2.2.2	Were there any unexpected costs related to getting seed from suppliers posted on the ICT tool?
C-Measures	a) ICT usage fees b) Costs to obtain seed (e.g. communication, transportation) c) Loans to pay for seed
C-Metric 2.2.3	Were there any changes in growing requirements for crops?
C-Measures	b) Changes in fertiliser, pesticides, labour or water requirements

	IMS: changes in growing requirements for seed obtained from suppliers posted on the ICT tool
Evaluation objective 2.2	*Assess changes in income*
C-Metric 2.2.1	Were there any changes in the yield obtained?
C-Measures	a) Yield/acre (kilograms) IMS: yield obtained with seed from suppliers posted on the ICT tool b) Incidence of crop diseases IMS: crop diseases with seed from suppliers posted on the ICT tool c) Incidence of loss due to pests IMS: pest problems with seed from suppliers posted on the ICT tool d) Farmers' rating of yield IMS: rating of yield with seed from suppliers posted on the ICT tool
C-Metric 2.2.2	Did the quality of the crop produced change?
C-Measures	a) Farmers' rating of crop quality IMS: rating of quality of crop grown with seed from suppliers posted on the ICT tool b) Yield obtained (C-Metric 2.2.1)
C-Metric 2.2.3	Were there any changes in the income farmers made from their crops?
C-Measures	a) Price/kilo obtained for crop IMS: price/kilo for crop grown with seed from suppliers posted on the ICT tool b) Yield obtained (C-Metric 2.2.1) c) Quality of crop (C-Metric 2.2.2) d) Change in total income from farming
Goal statement 3	**Delivering information on seed suppliers will change how farmers' choose a supplier and what seed they plant.**
Evaluation objective 3.1	*Assess changes in choice of seed*
C-Metric 3.1.1	Did the type of seed farmers plant change?
C-Measures	a) Change in seed variety IMS: variety presented by suppliers posted on the ICT tool b) Change in crop IMS: new crop presented by suppliers posted on the ICT tool
Evaluation objective 3.2	*Assess changes in choice of seed supplier*
C-Metric 3.2.1	Did farmers change their seed supplier?
C-Measures	a) Change in seed supplier IMS: farmers who bought seed from suppliers posted on the ICT tool
C-Metric 3.2.1	Where did the farmers get information regarding their choice of seed or seed supplier?

C-Measures	a) Based on their own past experience b) Talking with other farmers c) Seed supplier information via an ICT tool IMS: watched demonstration video IMS: created demonstration video IMS: participated in seed user group IMS: heard testimonials from other farmers IMS: gave their own testimonial IMS: participated in audio/video conference with supplier

The resulting measurement template can be used to evaluate the system's social impact (i.e. the impact of providing information on seed suppliers) based on the system's impact on the farmers' options regarding seed supplies, their profitability and the choices they actually make regarding seed suppliers and seed. Such a template can be used to identify baseline measurements to collect prior to introducing a system and after it has been deployed (in the case of seed, after a suitable time has passed to harvest and sell a crop). The same template can be used for a long-term assessment, or alternatively, to conduct a comparative study with respect to some other system related to seed suppliers (by defining appropriate implementation specific measures) or the case where no system is available. In this way, the goal models produced by the SDS process serve as the basis for defining a measurement framework for assessing a project's success based on its high-level social development goals rather than its technical success alone.

Chapter 9: Summary and Conclusion

The emergence of mobile phone communication and affordable ICT provide new opportunities for addressing critical social problems in the developing world. Realising this potential requires developing appropriate software applications to deliver relevant information in a manner whereby the intended audience can benefit from it. The nature of the targeted audiences and the social development problems addressed make this a challenging software design problem for which conventional requirements elicitation techniques are inadequate. The SDS methodology presented in this thesis is specifically designed to address the shortcomings conventional techniques present in an ICT4D context. In particular, it tackles the issues of inadequate attention paid to a project's high-level social development goals, neglect of environmental constraints, lack of input from end-users and disregard of social impact in project evaluation. The lack of participation by end-users in elicitation can be attributed to the difficulty that people of limited literacy have in articulating their problems and needs using conventional interview methods. Without adequate input from end-users regarding their specific needs and sociocultural context, it is difficult to ascertain the socioeconomic factors affecting a project's sustainable use and its ultimate ability to attain its stated social development goals. Assessing the extent to which a project achieves these goals is also essential to substantiate any claims of success and assure a project's long-term sustainability.

The methodology presented in this thesis draws on established theories and techniques from a range of disciplines, such as storytelling, communication theory, modelling in decision analysis and systems thinking, goal-oriented requirements engineering, requirements traceability and goal-based assessment. These techniques are combined in an original way to produce an approach that addresses the specific challenges of requirements engineering in the ICT4D domain. This thesis makes the following contributions to assist in gathering requirements for software systems that can bring real, sustainable benefits to the rural populations they target. Each of these contributions can be applied independently, and assembled together they constitute the building blocks of our proposed elicitation methodology.

1. **ICT4D quality model:** Based on our characterisation of ICT4D projects and the social and cultural aspects of technology acceptance and use, we have defined an ICT4D quality model in chapter 5, describing a set of desirable properties, or non-functional requirements, specific to ICT4D projects (P5, P8, P9, P10, P11, P16)[7]. Applying this model, software requirements

[7] These refer to the publications arising from our research, listed in appendix A.

analysts examine a project's feasibility, affordability and accessibility given the prevailing geographic and socioeconomic conditions and available skill set. A project's relevance, ability to inspire trust and potential benefit to the targeted community are likewise analysed. With regards to the dynamics of ICT use, a project's sustainability, support of a community of practice and cultural appropriateness are examined. Satisficing these properties will increase an ICT4D project's prospective acceptance, effective and sustainable use and scalability.

2. **Conceptual model of experiential culture:** In determining the cultural appropriateness of some ICT4D intervention, we have developed a model that is derived from Communication Theory, and Ong's theories on literacy in particular (Ong, 2002). Our model, based on the differences between experiential and analytic societies, provides a nuanced view of the deep aspects of culture that condition how people assimilate and use information in their daily lives (P1, P2, P3, P4, P5, P6, P15, P16). Based on the characteristics of an experiential society, we identify the following constraints related to communication and affecting change in a society: averse to disruptive change, reluctant to experiment, knowledge conveyed through concrete experience, not predisposed to abstract thinking, reluctant to act individually and high-context communication. The different viewpoints engendered by experiential versus analytic societies give rise to distinct forms of discourse, respectively characterised as "telling" versus "showing" (Taylor, 2008). Applying this model, we can determine the dominant cultural tendencies in a society and identify cultural constraints that might otherwise go unobserved and that can readily be mapped onto operational constraints and requirements for a proposed software based service.

3. **The SDS technique for eliciting needs from rural populations:** Achieving effective communication between software analysts and end-users from disadvantaged rural backgrounds using conventional elicitation techniques poses a major challenge due to the significant sociocultural differences between the two groups. Using the *Structured Digital Storytelling* technique whereby users express their needs through short "stories", we overcome the barriers of language, social class and literacy to elicit input from end-users directly, with minimal or no intervention from facilitators (P7, P12, P13, P14, P15). Starting with a set of themes or topics of interest, these are formulated into open-ended questions, translated and recorded in the local language, and then presented to users by means of an automated tool such as the E-Tool. The tool guides users through the elicitation process, playing the questions and recording answers. Once completed, the sequence of answers is

saved as a "story" in a library, where it can be accessed and played by other community members.

4. **The E-Tool—a prototype tool supporting SDS elicitation:** To support the SDS elicitation technique we have developed a prototype application called the E-Tool. Designed for use as a stand-alone application in a rural ICT4D setting, the tool factors the constraints of an experiential society and the principles embodied by our ICT4D quality model into its design (P12, P13, P14, P15). The tool is framed within a storytelling metaphor and offers users the option to (i) watch a video explaining the tool's purpose and operation, (ii) listen to stories told by other community members or (iii) tell their own story. To make it easy to use by a non-literate population the tool is equipped with a minimalistic, text-free user interface, which instead of text, uses graphical icons, buttons with distinct colors, and audio prompts in a structured dialogue to guide users through the various options. Planned enhancements include developing a version for collecting structured audio-stories using a mobile phone, with upload and download capabilities through the Internet to make the collection process more efficient and widely available.

5. **The SDS requirements elicitation methodology:** The SDS methodology applies the principles of user-driven requirements, goal-based analysis and requirements traceability to engineering software requirements for ICT4D. It augments the standard RE process by providing a systematic process for collecting, analysing and transforming information from end-users expressed as oral narratives into an analytic representation that can be linked to conventional RE processes (P12, P13, P14, P15). Given the high-level development goals, a set of topics related to the problem under investigation is selected. Using SDS elicitation on some appropriate tool, users are then asked to speak on those topics and their answers are collected in the form of digitally recorded stories. These narrations are then translated, analysed and abstracted in order to identify goals, needs, constraints and other relevant concepts within the problem domain. The analysis is enhanced by applying our model of experiential culture to identify cultural constraints that might not be obvious from simple observation. Applying our ICT4D quality model, additional goals, conditions and constraints may be introduced or alternatively eliminated. Using a goal-based analysis, the needs and constraints thus identified are integrated and elaborated until they converge on a single set of project requirements addressing the stakeholders' needs, including those of the end-users. The goal model and other artefacts derived from this analysis serve as primary input to a

conventional RE process and in deriving a measurement framework based on the project's high-level developmental goals.

Addressing the challenges of RE in the ICT4D domain: Our research has produced a methodology and set of supporting models, techniques and tools for eliciting requirements that addresses the main shortcomings conventional RE approaches present in a rural ICT4D context. With a goal-based approach, stakeholders are encouraged to explicitly state their goals and hence, include them in the analysis. Our ICT4D quality model assures that critical environmental constraints related to the sociodynamics of technology acceptance and use are factored into the analysis while our model of experiential culture assists in identifying cultural constraints based on cultural traits that are otherwise difficult to observe. Using the SDS elicitation technique, semi or non-literate users can express their needs in the form of stories, thus overcoming the barriers to participating in requirements gathering using conventional elicitation techniques. Through an incremental process of qualitative analysis and abstraction, contextual information expressed as oral narratives in a telling mode, is transformed into an analytic representation suitable for software analysts and readily integrated into conventional RE processes. Additionally, by linking the stories, goals and requirements, we provide traceability links between the 'raw' needs and software requirements. Moreover, with a measurement framework derived from the goal-model, we provide an instrument for assessing a project's success based on its social impact rather than its technical success alone.

An acceptable and effective methodology: The acceptability and effectiveness of the SDS elicitation technique and process for requirements elicitation have been validated using a proof-of-concept E-Tool in three field studies. These have demonstrated that SDS is both acceptable to end-users and effective in eliciting non-trivial contextual information regarding a project's purpose and software requirements. For example, in our farming study, insufficient funds and debt emerged as hard barriers preventing small farmers from being able to benefit from interventions that incur additional expenses, increase financial risk or involve a long term investment, with debt imposing an additional barrier on the ability to benefit from improved marketing conditions. Expanding the goal of "*providing information on seed suppliers*", we identified financial risk and distrust of suppliers as the primary obstacles preventing farmers from benefiting from such information. Drawing on our cultural model, we sought appropriate ways of conveying information to make it more acceptable and trustworthy to its intended audience, leading to the additional requirements of (i) providing positive concrete experience through observable results obtained by other farmers; (ii) supporting collective participation by the farmers and (iii) high-context intrapersonal communication with suppliers. The operationalisation of these requirements led to the identification of system features pertaining to playing and creating demonstration videos, supporting seed user groups and farmer testimonials, and providing video and/or audio-conferencing with suppliers in addition to delivering information on suppliers. Applying our ICT4D quality model, we identified the additional features of (iv) adding suppliers to the service; and (v) registering complaints and negative reviews of seed suppliers.

An adaptable and repeatable process: By conducting our experiment on different topics in different regions we have demonstrated that the SDS process is both adaptable and repeatable. Different topics can be addressed by changing the themes, and different linguistic regions are easily supported by simply translating and rerecording the prompts. As an elicitation technique, SDS demands comparatively few resources: preparatory work is minimal, skilled facilitators are not required, a tool is available, and data collection is relatively fast and accomplished at the participants' convenience. Because resources such as interviewers and facilitators are not involved, elicitation can be conducted more widely, increasing the number of stories collected and the coverage of issues. With enhancements, such as porting the elicitation application to a mobile phone platform and providing upload and download capabilities through the Internet, the collection process can be made more efficient and widely accessible to both participants and software analysts. Scaling up the translation and analysis to handle large collections of oral stories poses a different problem for which innovative approaches combining low cost man-power and automated machine capabilities are required.

Addressing the social development aspect of ICT4D: With the limited experiments conducted in the scope of this thesis, we have demonstrated that the SDS methodology appears eminently suitable for eliciting software requirements in an ICT4D context. It thus fills a critical gap that currently has no simple solution, namely how to get input from end-users regarding project goals and requirements. In doing so, it also brings ICT4D practice into better alignment with current social development theory that advocates involving local people in developing local solutions. Our ICT4D quality model, encapsulating the contextual factors pertaining to the sociodynamics of sustainable technology use in a rural context, encompasses the attributes of "life-enhancing offerings" promulgated by the World Economic Forum (2009). Additionally, by emphasising "support for a community of practice", our model aligns with social development through capacity building and empowerment, while its emphasis on "relevance" and "cultural appropriateness" promotes ICT as a positive, enabling tool. A storytelling approach leverages the customary mode of communication prevalent within experiential rural societies, with the process of telling and listening to stories having the potential benefit of raising the local population's awareness of the social problems they face and engaging them in the development effort (Kerr, 2003). With our model of experiential culture, we introduce a hitherto untapped branch of Communication Theory that is particularly suitable for examining cultural differences and social phenomena surrounding communication and affecting change in rural ICT4D contexts. Finally, by facilitating the assessment of social impact, our methodology promotes substantiating a project's developmental outcomes—an integral step in assuring a project's long-term sustainability and scalability.

The SDS methodology places a project's social development goals and related user needs upfront and foremost in driving requirements. Applying this methodology, analysts can obtain a more complete understanding of a problem domain and local conditions from the bottom-up, based on which they can prioritise the users' concerns, and identify and address the real-world obstacles preventing users from being able to fully benefit from some intervention. While this methodology in itself cannot guarantee that a project's outcome will be successful, its application will assure that many of the currently overlooked factors are considered in a design, improving a project's likelihood of success in terms of bringing real, sustainable benefits to its intended users.

Ideally, analysts will apply the SDS methodology in its entirety, to set an ICT4D project's vision, priorities and requirements at the elicitation stage. Alternatively, its constituent elements can be applied independently, to assess an existing project or to elaborate requirements for a new one, as we have done in designing the E-Tool. With regards to the SDS elicitation technique, this can be applied in the context of requirements elicitation, or for gathering other information, such as evaluation data or qualitative data in other fields where literacy, language or other barriers make conventional techniques difficult to apply. While SDS has interesting possibilities in the context of social science research, it is likely to require additional validation with respect to established techniques before it is generally accepted. Regarding the E-Tool itself, this can be deployed as a standalone mobile phone application. However, this presents certain logistic inconveniences regarding the installation of the application, distribution of phones and retrieval of stories. With appropriate enhancements, the E-Tool can be deployed as a mobile phone application connected via the Internet, increasing the efficiency and scalability of data collection considerably. As a future direction, we envision adapting the E-Tool concept to provide a tool for off-line communication and story sharing for purposes other than data collection.

References

Avgerou, C. (2001). The significance of context in information systems and organizational change, *Information Systems Journal*, 11(1): 43–63.

Babe, R. (2000). The communication thought of Harold Innis (1894–1952). In R. Babe (Ed.), *Canadian communication thought: Ten foundational writers*, 51–88. Toronto: University of Toronto Press.

Basili, V., Caldiera, G., Rombach, H. (1994). The Goal Question Metric Approach. In *Encyclopaedia of Software Engineering*, 528–532, John Wiley & Sons, Inc.

Beardon, H. (2006). ICT for Development: Empowerment or exploitation? In H. Rahman (Ed.), *Empowering marginal communities with information networking*. Hershey, PA: Idea Group Publishing, 44–61.

Berander, P., Jönsson, P. (2006). A Goal Question Metric Based Approach for Efficient Measurement Framework Definition, *International Symposium on Empirical Software Engineering (ISESE)*, 316–326.

Boehm, B. (2006). A view of 20th and 21st century software engineering. *International Conference on Software Engineering*, 12–29.

Brewer, E., Demmer, M., Du, B., Ho, M., Kam, M., Nedevschi, S., Pal, J., Patra, R., Surana, S., Fall, K. (2005). The case for technology in developing regions. *IEEE Computer*, 38(6), 25–38.

Bryl, V., Giorgini, P., Mylopoulos, J. (2009). Designing socio-technical systems: from stakeholder goals to social networks, *Requirements Engineering*, 14(1): 47–70.

Buhr, R. (1998). Use Case Maps as Architectural Entities, for Complex Systems, *IEEE Transactions on Software Engineering*, 24(12): 1131 – 1155.

Casebeer, A., Verhoef, M. (1997). Combining qualitative and quantitative research methods: Considering the possibilities for enhancing the study of chronic diseases, *Chronic Diseases in Canada (PHAC)*, 18(3), http://www.phac-aspc.gc.ca/publicat/cdic-mcc/18-3/d_e.html.

Chambers, Robert (1994). Participatory Rural Appraisal (PRA): Challenges, Potentials and Paradigm, *World Development*, 22(10): 1437–1454.

Cheng, B. H. C., Atlee, J. M. (2007). Research Directions in Requirements Engineering, *Proceedings of Workshop on the Future of Software Engineering* (ICSE'07), 285–303.

Couch, C. (1996). *Information Technologies and Social Orders*, Aldine de Gruyer.

Coughlan, J, Macredie R.D. (2002). Effective Communication in Requirements Elicitation: A Comparison of Methodologies, *Requirements Engineering*, 7(2): 47–60.

Chung, L., Nixon, B. A., Yu, E., Mylopoulos, J. (2000). *Non-Functional Requirements in Software Engineering*, Kluwer Academic Publishers.

Creswell, J. (2003). *Research Design: Qualitative, Quantitative and Mixed Methods Approaches*, 2nd ed., Sage Publications.

Dias, M., Brewer, E. (2009). How computer science serves the developing world, *Communications of the ACM*, 52(6): 74–80.

Dourish, P. (2006). Implications for design, *SIGCHI Conference on Human Factors in Computing Systems*, 541–550.

Esselink, B. (2003). The evolution of localization. *Guide to Localization*. Available at http://isg.urv.es/library/papers/Esselink_Evolution.pdf.

Forrester, J.W. (1994). System Dynamics, Systems Thinking and Soft OR, *System Dynamics Review*, 10(2–3): 245–256.

France, R., Rumpe, B. (2007). Model-driven Development of Complex Software: A Research Roadmap, *International Conference on Software Engineering (FOSE)*, 37–54.

Franklin, U. (1990). *The real world of technology*. Concord, Ont: Anansi Press.

Frohlich, D. M., Rachovides, D., Riga, K., Bhat, R., Frank, M., Edirisinghe, E., Wickramanayaka, D., Jones, M., Harwood, W. (2009). StoryBank: mobile digital storytelling in a development context, *International Conference on Human Factors in Computing Systems*, 1761–1770.

Garai, A. and B. Shadrach (2006). Taking ICT to Every Indian Village: Opportunities and Challenges, *One World South Asia*, New Delhi.

Gausepohl, K. A. (2008). *Investigation of Storytelling as a Requirements Elicitation Method for Medical Devices*. Masters Thesis in Industrial and Systems Engineering, Virginia Polytechnic Institute.

Gullivan, M., Srite, M. (2005). Information technology and culture: Identifying fragmentary and holistic perspectives of culture, *Information and Organization*, 15(4): 295–338.

Hands, K., Peiris, D.R., Gregor, P. (2004). Development of a computer-based interviewing tool to enhance the requirements gathering process, *Requirements Engineering*, 9(3): 204–216.

Havelock, E.A. (1963). *Preface to Plato*, Cambridge MA: Belknap/Harvard University Press.

Heeks, R. (2002). Information Systems and Developing Countries: Failure, Success, and Local Improvisations, *The Information Society*, 18(2):101–112.

Heeks, R. (2008). ICT4D 2.0: The Next Phase of Applying ICT for International Development, *IEEE Computer*, 41(6): 26–33.

Hirsch, G.B., Levine, R., Miller, R.L. (2007). Using system dynamics modeling to understand the impact of social change initiatives, *American Journal of Community Psychology*, 39(3–4): 239–253.

Ho, M., Smyth, T., Kam, M., Dearden, A. (2009). Human-Computer Interaction for Development: The Past, Present, and Future, *Information Technologies and International Development*, 5(4): 1–18.

Hofstede, G. (2005). *Cultures and organizations: Software of the mind.* New York: McGraw-Hill.

Howard, R.A., Matheson, J.E. (2005). Influence Diagrams, *Decision Analysis*, 2(3): 127–143.

Huitt, W. (2004). Maslow's hierarchy of needs. *Educational Psychology Interactive.* Valdosta, GA: Valdosta State University. Retrieved July 24th, 2006 http://chiron.valdosta.edu/whuitt /col/regsys/maslow.html

IEEE standard 1233-1998. *IEEE Guide for Developing System Requirements Specifications*, Institute of Electrical and Electronics Engineers, Inc. New York

International Organization for Standardization (ISO) (2001). *ISO 9126-1 Software engineering: Product quality, Part 1, Quality model*, ISO/IEC, Switzerland.

Johnson, R., Onwuegbuzie, A. (2004). Mixed Methods Research: A Research Paradigm Whose Time Has Come, *Educational Researcher*, 33(7): 14–26.

Jones, S., Maiden, N. (2005). RESCUE: An integrated method for specifying requirements for complex sociotechnical systems. In J. Maté & A. Silva (Eds.), *Requirements Engineering for Sociotechnical Systems*, 245–265. Hershey, PA: Information Sciences Publishing.

Kerr, D. (2003). "We Know What the Problem Is": Using Oral History to Develop a Collaborative Analysis of Homelessness from the Bottom Up, *The Oral History Review*, 30(1): 27–45.

Kersten, G. E., Kersten, M. A., Rakowski, W. M. (2002). Software and culture: Beyond the internationalization of the interface, *Journal of Global Information Management*, 10(4): 86–101.

Kleine, D., Unwin, T. (2009). Technological revolution, evolution and new dependencies: what's new about ICT4D? *Third World Quarterly*, 30(5): 1045–1067.

Knack, S., Rahman, A. (2007). Donor Fragmentation and Bureaucratic Quality in Aid Recipients, *Journal of Development Economics*, 83(1): 176–197.

Koch, J. L., Caradonna, T. M. (2006). Technologies and business models that work in developing countries, *International Conference on Information and Communication Technologies and Development*, 193–202.

Larman, C. (2005). *Applying UML and patterns: an introduction to object-oriented analysis and design and iterative development,* 3rd ed. Prentice Hall PTR.

Lave, J., Wenger, E. (1991). *Situated Learning: Legitimate Peripheral Participation.* Cambridge University Press.

Leffingwell, D., Widrig, D. (2003). *Managing software requirements: a use case approach*, Addison-Wesley, Boston.

Matulevičius R., Heymans P., Guttorm , S. (2006). Comparing goal-modelling tools with the RE-tool evaluation approach, *Information Technology and Control*, 35(3A), 276–286.

Medhi, I., Sagar, A., Toyama, K. (2007). Text-Free User Interfaces for Illiterate and Semi-Literate Users, *Information Technologies and International Development*, 4(1): 37–50.

Medhi, I., Toyama K. (2007). Full-Context Videos for First-Time, Non-Literate PC Users, *International Conference on Information and Communication Technologies and Development*, Bangalore, India.

Millen, David. R. (2000). Rapid Ethnography: Time Deepening Strategies for HCI Field Research, *Designing Interactive Systems: processes, practices, methods, and techniques*, 280–286.

Mingers, J. (2001). Combining IS Research Methods: Towards a Pluralist Methodology, *Information Systems Research*, 12(3): 240–259.

Mohan, G. (2006). Beyond participation: strategies for deeper empowerment. In Cooke, B. and Kothari, U. (Eds.), *Participation: the new tyranny?* London: Zed Books, 153–167.

Myers, M. D. (1997). Qualitative Research in Information Systems, *MIS Quarterly*, 21(2) 241–242. *MISQ Discovery*, updated October 1, 2009, www.qual.auckland.ac.nz.

Myers, M. D., Tan, F. B. (2003). Beyond models of national culture in information systems research. *Advanced topics in global information management*, IGI Publishing, 14–29.

Nuseibeh, B., Easterbrook, S. (2000). Requirements Engineering: A Roadmap, *The Future of Software Engineering (FOSE)*, companion volume to the *22nd International Conference on Software Engineering* (ICSE'00), 35–46.

Ong, Walter J. (2002). *Orality and literacy: The technologizing of the word*. New York: Routledge.

Orbuch, T. (1997). People's Accounts Count: The Sociology of Accounts, *Annual Review on Sociology*, 23(1): 455–478.

Ottens, M., Franssen, M., Kroes, P. A., Poel, I. v.d. (2005). Systems Engineering of Socio-Technical Systems, *INCOSE International Symposium*, Rochester, NY.

Pearl, J. (2000). The art and science of cause and effect. *Causality*, Cambridge University Press, 331–379.

Pitula, K. (2007). A Framework for Inclusive Technology: Incorporating Development Goals into ICT Design, *COGNITIO Conference in Cognitive Science*, Montreal.

Pitula, K. (2009). India E-Tool data analysis: Farmers' narrations, Technical Report, Concordia University, Montreal.

Pitula, K., Dysart-Gale, D., Radhakrishnan, T. (2010). Expanding the Boundaries of HCI: A case study in requirements engineering for ICT4D, *Journal of Information Technologies & International Development*. 6(1): 78–93.

Pitula K., Radhakrishnan T. (2007a). A conceptual model of inclusive technology for Information access by the rural sector, *Proceedings of the International Conference on Human-Computer Interaction (HCII'07)*, 243–252.

Pitula K., Radhakrishnan T. (2007b). A set of heuristic measurements for evaluating the inclusiveness of a technology, *Proceedings of the Home Informatics and Telematics (HOIT) Conference*, 35–48.

Pitula, K., Sinnig, D., Radhakrishnan, T. (In press). Requirements Engineering in the ICT4D Domain. In A.G. Abdel-wahab, A. El-Masry (Eds.), *Mobile Information Communication Technologies Adoption in Developing Countries: Effects and Implications*; Published by IGI Global (www.igi-global.com).

Potts Steves, M., Scholtz, J. (2005). A Framework for Evaluating Collaborative Systems in the Real World, *Hawaii International Conference on System Sciences*, 29c.

Prakash, A., De', R. (2007). Importance of development context in ICT4D projects: A study of computerization of land records in India. *Information Technology and People*, 20(3): 262–281.

Pressman, R. S. (2005). *Software engineering: A practitioner's approach*, McGraw-Hill, Boston, Mass.

Ramirez, R. (2001). A model for rural and remote information and communication technologies: a Canadian exploration. *Telecommunications Policy*, *25*(5), 315–330.

Rashid, A., Elder, L. (2009). Mobile phones and development: an analysis of IDRC-supported projects, *Electronic Journal of Information Systems in Developing Countries*, 36(2): 1–16.

Rational Software Corporation. Retrieved May 4, 2007, http://www.ts.mah.se/RUP/RationalUnifiedProcess/

Reimer, W. (2005) Rural and Urban: Differences and Common Ground, in *Urban Canada: Sociological Perspectives*, ed. Harry H. Hiller, Oxford University Press.

Rezwan (2009). India: A wave of suicides among farmers, *Global Voices Online,* August 25. Retrieved September 12, 2009 at http://globalvoicesonline.org/2009/08/25/india-a-wave-of-suicides-among-farmers/

Rice, M. (2003). Information and Communication Technologies and the Global Digital Divide: Technology Transfer, Development, and Least Developing Countries, *Comparative Technology Transfer and Society*, 1(1): 72–88.

Robinson/Sunna, S. (2007). Seeds of Despair, *Time.com*, May 17. Accessed 09/12/2009 http://www.time.com/time/magazine/article/0,9171,1622180,00.html.

Rogers, Y. (2004). New Theoretical approaches for Human-Computer Interaction, *Annual Review of Information, Science and Technology*, 38: 87–143.

Rolland, C., Proix, C. (1992). A Natural Language Approach for Requirements Engineering, *Proceedings of CAiSE 1992 on Advanced Information Systems Engineering*, Manchester, UK.

Saiedian, H., Dale, R. (2000). Requirements engineering: making the connection between the software developer and customer, *Information and Software Technology*, 42(3): 419–428.

Scholtz , J., Potts Steves, M. (2004). A Framework for Real-World Software System Evaluations, *ACM Conference on Computer Supported Cooperative Work* (CSCW), 600–603.

Snowden, D. (1999). Story telling: an old skill in a new context, *Business Information Review*, 16 (1), pp. 30–37.

Sterman, J.D. (2001).System Dynamics Modeling: Tools for Learning in a Complex World, *California Management Review*, 43(4): pp.8–25.

Supakkul, S. (2010) RE-Tool, http://www.utdallas.edu/~supakkul/tools/RE-Tools/index.htm, retrieved January 26, 2010.

Suttcliffe, A. (2000). Requirements analysis for socio-technical system design, *Information Systems*, 25(3): 213–233.

Taylor, A. (2000). IT Projects: Sink or Swim?, *The Computer Bulletin*, 41 (1): pp. 24–26.

Taylor, C. (2008). Trafficking in facts: Writing practices in social work. *Qualitative Social Work*, 7(1): 25–42.

Tongia, R., Subrahmanian, E. (2006). Information and Communications Technology for Development (ICT4D) – A Design Challenge? *International Conference on Information and Communication Technologies and Development*, 243–255.

Trainer, T. (2002). Development, charity and poverty: the appropriate development perspective, *International Journal of Social Economics*, 29(1/2): 54–72.

Trochim, W. M. (2006). *The Research Methods Knowledge Base*, 2nd Edition. Internet WWW page. Retrieved January 26, 2010 at URL: http://www.socialresearchmethods.net/kb/ (version current as of October 20, 2006).

United Nations (2009). *The Millennium Development Goals Report.* Retrieved January 12, 2009, http://www.un.org/millenniumgoals/pdf/MDG_Report_2009_ENG.pdf

Unwin, T. (2009). *ICT4D: Information and communication technology for development.* London: Cambridge University Press.

Van Lamsweerde, A. (2001). Goal-Oriented Requirements Engineering: A Guided Tour, *IEEE International Symposium on Requirements Engineering*, 149–163.

Venkatesh, V., Morris, M.G., Davis, G.B., Davis, F.D. (2003). User Acceptance Of Information Technology: Toward A Unified View, *MIS Quarterly*, 27(3): 425–478.

Warschauer, M. (2003). *Technology and Social Inclusion*, MIT Press.

Williams, A., Dourish, P. (2006). Imagining the City: the Cultural Dimensions of Urban Computing, *Computer*, 39(9): 38–43.

World Economic Forum (2009). *The Next Billions: Unleashing Business Potential in Untapped Markets*, Report REF: 150109, Switzerland.

Appendix A: List of publications ensuing from this research

9.1 Journal publications

P1: Dysart-Gale, D., **Pitula, K.**, Radhakrishnan, T. (2011). Culture, Communication and ICT for Development: A Caribbean Study, *IEEE Transactions on Professional Communication*, March issue.

P2: Dysart-Gale, D., **Pitula, K.**, Radhakrishnan, T. (2010). Improving Professional Writing for Lay Practitioners: A Rhetorical Approach, *IEEE Transactions on Professional Communication*, December issue.

P3: **Pitula, K.**, Dysart-Gale, D., Radhakrishnan, T. (2010). Expanding the Boundaries of HCI: A case study in requirements engineering for ICT4D, *Journal of Information Technologies & International Development,* 6(1): 78–93.

P4: Dysart-Gale, D., **Pitula, K.**, Radhakrishnan, T. (2009). Reports and recommendation writing for development: Rhetorical skills for social action, *IEEE Technical Communication*, 56(4), 387–396.

9.2 Conference proceedings

P5: **Pitula, K.**, Sinnig, D., Radhakrishnan, T. (2010). Making Technology Fit: Designing an Information Management System for Monitoring Social Protection Programmes in St. Kitts, *Sir Arthur Lewis Institute of Social & Economic Studies Annual Conference (SALISES)*, University of the West Indies, Trinidad & Tobago.

P6: Dysart-Gale, D., **Pitula, K.**, Radhakrishnan, T. (2009). A community-driven communicative approach to adoption of a client record management system, *Proceedings of the Pan American Health Care Exchanges Conference* (PAHCE, endorsed by IEEE, International Federation for Medical and Biological Engineering (IFMBE), and World Heath Organization (WHO), among others), 50–52.

P7: **Pitula, K.**, Radhakrishnan, T. (2008). A multimedia tool to elicit information needs in rural communities, *Workshop on HCI for Community and International Development*, 26th *Conference on Computer Human Interaction* (CHI'08), Florence, Italy.

P8: **Pitula, K.**, Radhakrishnan, T. (2007). A framework and process for designing inclusive technology, *International Conference on Software Engineering Advances* (ICSEA 2007), Cap Esterel, France, 64.

P9: **Pitula K.**, Radhakrishnan T. (2007). A conceptual model of inclusive technology for Information access by the rural sector, *Proceedings of the International Conference on Human-Computer Interaction (HCII'07)*, 243–252.

P10: **Pitula K.**, Radhakrishnan T. (2007). A set of heuristic measurements for evaluating the inclusiveness of a technology, *Proceedings of the Home Informatics and Telematics (HOIT) Conference*, 35–48.

P11: **Pitula, K.** (2007). A Framework for Inclusive Technology: Incorporating Development Goals into ICT Design, *COGNITIO Conference in Cognitive Science.*

9.3 Books

P12: **Pitula, K.**, Sinnig, D., Radhakrishnan, T. (In press). Requirements Engineering in the ICT4D Domain. Book chapter in A. El-Masry (Ed.), *Mobile Information Communication Technologies Adoption in Developing Countries: Effects and Implications*; Published by IGI Global (www.igi-global.com).

9.4 Submitted

P13: **Pitula, K.**, Radhakrishnan, T. (in review). On eliciting requirements from end-users in the ICT4D Domain. Manuscript submitted January 2010, *Journal of Requirements Engineering.*

P14: Sinnig, D., **Pitula, K.**, Becker, R., Radhakrishnan, T., Forbrig, P. (in review). Structured Digital Storytelling for Eliciting Software Requirements in the ICT4D Domain, *International Conference on Human-Computer Interaction (HCII'10).*

9.5 In progress

P15: **Pitula, K.**, Radhakrishnan, T. (in progress). A storytelling approach for participatory requirements gathering in rural ICT4D contexts, *1st ACM Annual Symposium on Computing for Development* (DEV 2010, co-located with ICTD 2010, hosted and sponsored by UNESCO, IDRC, and Microsoft among others).

P16: **Pitula, K.**, Sinnig, D., Radhakrishnan, T. (in progress). Introducing technology in a small island context: eliciting requirements for an Information Management System on St. Kitts. Book chapter in D. Dysart-Gale (Ed.), *Small Islands, Big Questions: A Multidisciplinary Approach to Problems of Caribbean Development.*