ITB Journal

Issue Number 7, May 2003

Contents

Papers for submission to the next ITB Journal should be sent to the editor at the address below. Alternatively, papers can be submitted in MS-Word format via email to brian.nolan@itb.ie

Brian Nolan
Editor
ITB Journal
Institute of Technology Blanchardstown
Blanchardstown Road North
Blanchardstown
Dublin 15

Editorial

I am delighted to introduce the seventh edition of the ITB Journal, the academic journal of the Institute of Technology Blanchardstown. The aim and purpose of the journal is to provide a forum whereby the members of ITB, visitors and guest contributors from other third level colleges can publish an article on their research in a multidisciplinary journal. The hope is that by offering the chance to bring their work out of their specialised area into a wider forum, they will share their work with the broader community at ITB and other academic institutions.

In this issue we have once again a diverse mix of interesting papers from across a range of different disciplines. We have a paper on Unified Internet Messaging from Paul Healy and Declan Barber, School of Informatics and Engineering. ITB. This is one of a series exploring different aspects of unified messaging. From the School of Business and Humanities, ITB we have a detailed paper on The Process and Problems of Business Start-ups by Natasha Evers, while from the computer science department of IT Tallaght we have a paper Eamonn Hyland and Dean Fennell that outlines some experiences gained from the deployment of an E-Learning "Java Arrays" Prototype for novice Java programmers in that college. We have a paper on Hardware/Software Codesign from Richard Gallery and Deepesh M. Shakya, School of Informatics and Engineering, ITB in which they outline work being undertaken as part of a research project. An interesting collaborative paper from S.D. McLoughlin, School of Informatics and Engineering, ITB and C. O'Rourke, J. McDonald, C.E. Markham, all from the Dept. of Computer Science, in NUI Maynooth discusses the Integration of a Stereo Vision System and GPS Data for Recording the Position of Feature Points in a Fixed World Coordinate System. Finally, from Mark Cummins, School of Informatics and Engineering, ITB we have a paper on a Secure Payment Protocol within an iBank context.

Once again, we hope that you enjoy the papers in this issue of the ITB Journal.

Brian Nolan
Editor
ITB Journal
Institute of Technology Blanchardstown
Blanchardstown Road North
Blanchardstown
Dublin 15

Unified Internet Messaging

Paul Healy and Declan Barber,

Institute of Technology, Blanchardstown

Paul.Healy@itb.ie

Declan.Barber@itb.ie

Abstract

As telephony services, mobile services and internet services continue to converge, the prospect of providing Unified Messaging and even Unified Communications becomes increasingly achievable. This paper discusses the growing importance of IP-based networks to Unified Messaging developments and examines some of the key services and protocols that are likely to make Unified Messaging more widely available. In this initial paper, we limit ourselves initially to the unification of text-based messaging using SMS and Email. The approach we make is based on the existing Internet Email framework but will take cognisance of the need to add voice and other non-text based messaging and communications at a later stage. Ultimately, the research project aims to build a working prototype of a generic messaging model that works for both non real-time and real-time communications. This model will provide a framework into which existing and future messaging protocols can be plugged. This paper originated from an applied research project which examined the integration possibilities for various messaging and communications protocols

1. Introduction

As traditional telephony services, mobile services and Internet services continue to converge, the prospect of providing true Unified Messaging (UM) and even Unified Communications (UC) becomes increasingly achievable. For the purposes of this paper we will define Messaging as any non-real time or near-real time message transfer service (e.g. email, paging, sms , fax, voicemail) and Communications as any real time, near-real time or non-real time messaging service (e.g. email, sms, fax, voicemail, telephony, computer telephony, video-conferencing). We define the minimum set of features in a truly Unified System (US) to include:

- a unified inbox for each user/profile
- a unified (logical) message store
- a unified directory service
- unified management of either centralised or distributed component architecture
- conversion between different media
- universal access from a broad range of user devices.

The ability to deliver a broad range of services over the ubiquitous Internet Protocol (IP) is driving the convergence of services traditionally provided over voice, data and Internet technologies. The UM of the future will need to be able to support real-time and non real-time messaging and communications based on a variety of media (text, voice, video and indeed multimedia). Traditionally, IP networks have not supported mixed traffic, but protocols are now emerging that can address the class of service issues for different types of traffic. There is widespread support in IP-based networks for email protocols and related extensions for text, voice, graphic and video attachments. Voice has not yet been implemented to the same extent using open standards over IP, but that is rapidly changing. Before we move on to examining specific Internet protocols, it is worth examining the benefits of an IP network.

2. Networks Converge towards IP

A review of the more significant differences between traditional network types is informative in understanding the benefits of using IP-based communications. A brief overview of the relative merits of traditional voice, data and IP networks is shown in Table 1. From the table, it is clear that there are numerous advantages to using IP networks, not least of which is the ability to control bandwidth. As IP protocols continually evolve to support features such as Quality of Service and Traffic Engineering, IP is reaching a stage where it may begin to offer the combination of the quality and reliability of the traditional circuit-switched network with the scalability and flexibility of packet-switched networks.

Arising from these benefits, the telecommunications world is making a shift away from traditional circuit-switched technologies towards a more open packet-switched infrastructure based on routed IP networks. This does not mean that the Internet will replace the traditional telephone and data networks in the foreseeable future but that the convergence of these technologies will lead to increased interoperability and consequently enhanced services. Consequently, our work is firmly focused on IP as the core set of protocols around which we will build our Unified Messaging prototype.

The earliest approaches to unified messaging used proprietary clients which integrated messages in a single application from a number of different message servers, each of which used its own message store and directory. This approach has been superseded by the dominance of clients based on proprietary email systems, such as Microsoft Exchange, with powerful Application Programming Interfaces (APIs). If we consider internet-based unified

messaging, there is the option to use a web-browser based client or widely available open standard email client that conforms to standard email protocols identified in section 5 below. Even the main proprietary email clients now support these standard protocols. Browser based access is server-centric with the web server acting as a proxy client to the end user who views the result in a HTML page. While this may initially provide a degraded email functionality, it does provide the most open, universal access.

	Voice Networks	**Data Networks**	**IP-Based Networks**
Availability	Very High	Congestion possible	Very High
Reliability	End-to-End	Only within the provider network	End-to-End
Scalability	Moderate	Moderate	Very High
Adaptability /Integration	Low	Low	High
Flexibility	Low	Variable	High
Ability to Control Bandwidth	Low	Low	High
Switching	Circuit-Switched	Packet-Switched	Packet-Switched
Standards	Standards based Equipment and software may be proprietary	Open Standards Based. Interoperable.	Open and Highly interoperable with increasing availability of APIs
Features /Services	Well-developed feature set	Variety of services	Supports multiple services and rich features
Connection	Dedicated and guaranteed Constant Bit Rate	Various connection types possible	Can support both Connection-oriented and Connectionless with varying classes of service.
Relative Ability to withstand Errors	High	Low	Variable
Delay	Low Deterministic	Variable	Traditionally variable. Emerging Protocols (MPLS) make it more deterministic

Table 1: An overview of the relative merits of traditional networks

3. Internet Email - Paradigms

There are a number of different approaches to building a distributed email infrastructure.

- Shared file-system strategies

- Proprietary LAN-based protocols
- X.400 P7 protocol
- Internet message access protocols

The only relevant approach for this project is Internet message access protocols: POP (Post Office Protocol), DMSP (Distributed Mail System Protocol), and IMAP (Internet Message Access Protocol). Of the three, POP is the oldest and consequently the best known. DMSP is largely limited to a single application, PCMAIL is known primarily for its excellent support of "disconnected" operation. IMAP offers a superset of POP and DMSP capabilities, and provides good support for all three modes of remote mailbox access: offline, online, and disconnected.

In the **Offline Paradigm,** mail is delivered to a mail server, and a personal computer user periodically invokes a mail "client" program (e.g., Outlook Express) that connects to the server and downloads all of the pending mail to the user's own machine. Thereafter, all mail processing is local to the client machine. The offline access mode is a kind of store-and-forward service, intended to move mail on demand from the mail server to a single destination workstation. Once delivered to the workstation, the messages are then deleted from the mail server and the user can continue to view and manipulate the messages even while offline.

In the **Online Paradigm** mail is again delivered to a shared server, but the mail client does not copy it all at once and then delete it from the server. This is more of an interactive client-server model, where the client can ask the server for headers, or the bodies of specified messages, or to search for messages meeting certain criteria. Messages in the mail repository can be marked with various status flags (e.g. "deleted" or "answered") and they stay in the repository until explicitly removed by the user, which may not be until a later session.

In the **Disconnected Paradigm**, the disconnected access mode is a hybrid of the offline and online models, and is used by protocols such as PCMAIL. In this model, a client user downloads some set of messages from the server, manipulates them offline, then at some later time uploads the changes. The server remains the authoritative repository of the messages. The problems of synchronisation arising from this mode (particularly when multiple clients are involved) are handled through the means of unique identifiers for each message.

4. Internet Email – Architecture

In Internet based email, the protocols used are highly dependent on the underlying Internet architecture. There is widespread support in IP-based networks for the Simple Mail Transfer

Protocol (SMTP) and the Multi-Purpose Mail Extension (MIME) for text, voice, graphic and video support. These protocols are designed to run over a client/server architecture, where there is an email server component with which the client programs communicate via direct connections such as sockets or remote procedures calls. Internet Email operates using a store and forward process. Both the client and the server must have an appropriate transport layer and a network API in place e.g. TCP/IP, which can be used by a developer to implement the protocols. The message store (or database) needs of an email system are generally simpler than those provided by a relational database and is only accessible via the server program. The store should be transparent to other programs. The email client is typically single-threaded and is only run when needed. It only needs to maintain a single connection to an email server. The server, in contrast, must be running at all times and starts automatically when the server is booted up. It must be capable of supporting multiple connections simultaneously from other clients and servers. The design should be both multitasking and even multithreaded. In such a client/server architecture, the intensive processing, such as for a search, is conducted on the server and only the result is sent back to the client. This minimises the amount of network I/O needed. Client/server are distributed architectures and are generally scalable to millions of users.

5. Internet Email – Protocols

To send mail, the email client (or user agent) establishes a network connection to a server and once connected, uses the Simple Mail Transfer Protocol (SMTP) to handle outgoing messages. To retrieve email, a client uses either the Post Office Protocol Version 3 (POP3) or more efficiently, the Internet Message Access Protocol, Version 4 (IMAP4). It is not necessary for the sender and receiver clients to be using the same retrieval protocol. The internal component in the server that collects and distributes the mail to different servers is called the Message Transfer Agent (MTA). It usually runs two main processes: an SMTP process to receive emails from a client and to forward emails to a different email server, and a POP3 or IMAP4 process to support the retrieval of mail. These processes may be run on the same or on different servers. The message store on the server is a specialised free-form textual database. This may be implemented using real database managers or use some internal database functionality. Apart from storing the messages, the MTA may record other useful data, such as if the message has been read or not (although strictly speaking, this is part of the clients responsibility). The message store may be implemented on a separate machine from the MTA or even across a number of different machines. It is utterly transparent form

the client. Message queues, such as an outgoing message queue or failed delivery queue are used to manage messaging. The Multipurpose Internet Mail Extension (MIME) to SMTP allows various kinds of files such a document, image or audio files, to be attached to text mail. It achieves this using binary-to-text encoding. As well as encoding, MIME adds data to the message header, so it is not a process that can be executed in advance of client activation. Rather, to send a MIME attachment, MIME must be integrated into the client. MIME facilitates the insertion of intelligence into the attachment recovery process by identifying the type of attachment to be specified (e.g. audio or graphic). MIME also supports the organisation of messages in various ways (e.g. in parallel or in multiple parts). Although not obliged to, email servers usually process at least some of the MIME header information to optimise performance. MIME is quite complex and as a result, most clients implement only a reasonably small part of the standard.

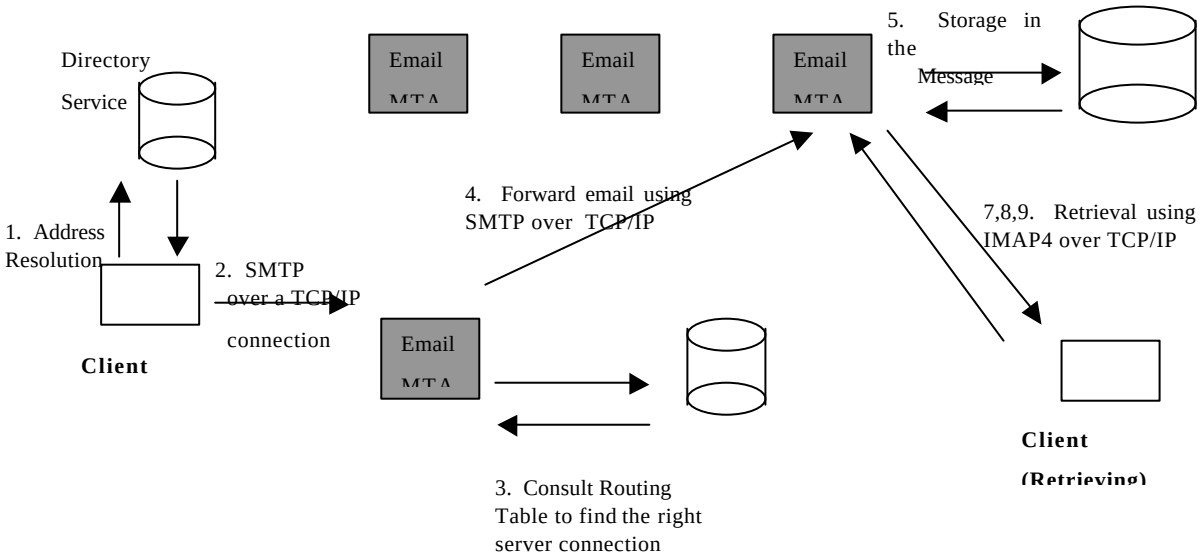


Figure 1: Internet Email Architecture and Operation

6. Comparison of POP3 and IMAP4

POP is the most commonly used Internet mail protocol. This protocol is the easiest to implement and use. Until a few years ago mail servers and client readers only supported POP. POP is still the default protocol for most mail reader clients and is supported by products on PCs Macs and Unix systems. POP is based entirely on the offline paradigm and therefore suffers limitations. It is only suitable for use with one client machine per account because messages are deleted from the server once they are downloaded to a machine. Once a connection is established with POP, all messages and their entire messages bodies, including

attachments, are downloaded to the client application. This creates a time and bandwidth overhead for unwanted messages. However POP is very easy to implement from both the client and server ends and is widely supported. POP is most suitable for home users that use just one machine for mail delivery.

IMAP is based on the online and disconnected paradigms where all the mailbox and messages are maintained on the server. It may also be set up for offline use if desired. IMAP allows the client to access and manipulate remote messages as if they were stored locally. The client issues commands to download them or delete them, access and set message state information, but the server always maintains the information. IMAP allows for faster service because the client reads all of the message headers without having to download all of the actual messages. For example, if a mailbox has ten messages in it and two of those messages has 200kb attachments then obviously it would be preferable to view the headers for these messages first and then decide if it is worth spending extra time and bandwidth downloading the files.

IMAP vs. POP:

Advantages of POP:

- Simple protocol to implement.
- More supporting client software available.

Advantages of IMAP:

- Can manipulate persistent message status flags.
- Stores messages as well as fetch them.
- Support for multiple mailbox management.
- Support for concurrent updates and access to shared mailboxes.
- Suitable for accessing non-email data; e.g., NetNews, documents.
- Also suitable for offline paradigm, for minimum connection time and disk use.
- Online performance optimisation, especially over low-speed links.

POP and IMAP can basically be compared based on the suitability of the supporting message access paradigms. So while offline and online mailers both allow access to new incoming messages on the mail server from a variety of different client platforms, the similarities stop there. The two paradigms reflect different requirements and styles of use and they do not mix very well. Offline works best for people who use a single client machine all the time. It is not well-suited for the goals of accessing an inbox of recent messages or saved-message folders from different machines at different times. This is because the use of offline ("download and delete") mail access from different computers at different times, tends to scatter the mail

across the different computers, unless they are all linked to a common network file system (in which case the access mode is really more online than offline.) On the other hand, the chief virtue of offline access is that it minimises the use of server resources and connect time when used via dialup.

IMAP (online) is the preferred access method for web-browsers because it only downloads the headers, whereas POP (offline) will store the entire messages in the browser cache. This can lead to out-dated files being kept on the local machine even after the user has deleted them from their main local message store.

7. Internet Directory Services

The ability to map a list of people/names to a relevant address is a critical part of the Internet structure. This need is analogous to the need for a telephone directory for PSTN numbers. In relation to email, we can consider this address mapping at locations: in the client, in the enterprise and globally. Proprietary email systems have developed directory services using a centralised approach to support all but the latter extremely well. The Internet has traditionally been weak in this area. A strong standard, X.500, was developed by the ISO to keep track of email addresses for X.400 mail but it requires the use of an ISO network stack and is therefore not appropriate for the Internet. A standard called the Lightweight Directory Access Protocol (LDAP), which is based on part of the X.500 standard but which runs on a TCP/IP stack, was developed for use on the Internet to manage internet-based email addresses. Most email clients, both proprietary and Internet–based, now support LDAP and a global network of LDAP servers and databases can be developed to facilitate finding someone's email address.

8. Short Message Service (SMS)

Developed in 1991, SMS (more commonly known as Text Messaging) is a globally accepted wireless service that enables the transmission of alphanumeric messages between mobile subscribers and external systems like email, paging, and voice mail systems. With SMS, an active mobile handset can receive or submit a short message at any time, even if a voice or data call is in progress. SMS also guarantees delivery of the short message by the network. Temporary failures due to unavailable receiving stations are identified, and the short message is stored until the destination device becomes available.

Initial applications of SMS focused on eliminating alphanumeric pagers by permitting two-way general-purpose messaging and notification services, primarily for voice mail. As technology and networks evolved, a variety of services were introduced, including interactive banking, information services such as stock quotes, integration with Internet-based applications, and email, fax, and paging integration. In addition, integration with the Internet spurred the development of Web-based messaging and other interactive applications such as instant messaging, gaming, and chatting.

SMS is currently more popular with users than the WAP protocol. It is also a simpler protocol for which to create an application. The following are some reasons why SMS is more suitable than WAP for value-added services:

- Large number of legacy (non-WAP enabled) phones
- WAPs uncertain future
- Lack of widespread WAP content
- SMS is suitable for meeting major market needs
- SMS can be used as a kind of 'push technology', meaning the user does not have to request delivery of information. It can be automatically sent.

9. SMS Connectivity:

This project will submit SMS messages from web-browsers. There are three distinct approaches to this:

- To **send an SMS from an attached GSM modem** over the normal air interface. This has the advantage of being able to send and receive SMS messages directly but is not very scalable and is disregarded for that reason.

- **HTTP Post Method:** to simply post the message data from the browser directly to the project web server. The web server will then forward this data to the core SMSC, where it will be processed and sent on to the appropriate mobile number(s).

- **TCP/IP Connection**: to post the message data to the project web server where a server-side application will establish a TCP/IP connection with the SMSC and transmit the message over IP.

The **HTTP method** of SMS communication allows for one-way transmission of single or multiple SMS messages from a web browser to mobile handsets. This is the limit of

functionality of this method. The browser on the client machine downloads the web page from a local or remote server, fills in the message and recipient details and then submits the message details as an HTTP form. However the form details are not submitted back to the server from where the web-page was downloaded. Instead it posts the message data to a third-party SMS handler, which then forwards the message data on to the SMSC. The SMSC is responsible for propagating the message(s) on to the recipient(s). This is usually done for some nominal fee.

Figure 2 depicts the process described above. The single-way arrows connecting the client machine, third-party server and the SMSC depict the communication limitation to this approach. The only return from a message submission is either confirmation of receipt at the SMSC or one of a host of possible error codes. The third-party server and the SMSC may in some instances belong to the same organisation. This does not affect the message submission process.

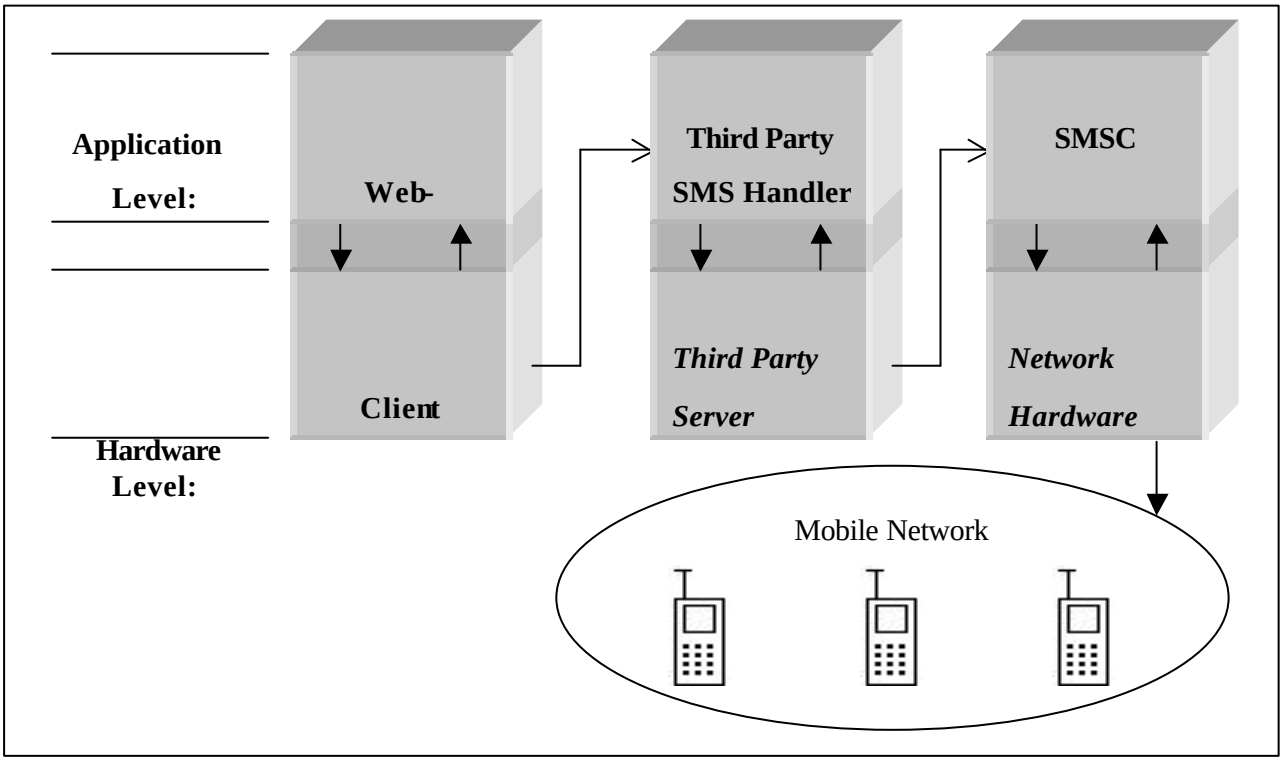


Figure 2: Diagram of HTTP Post Method

TCP/IP Transmit Method:

The TCP/IP method requires a program running on the project server which can access the SMSC directly without using a third party. The web page is downloaded into users browser from the project server, the message details are filled out as before but this time they are

posted back to the project server from which the page was downloaded. A program running on the server will then establish a TCP/IP connection (possibly synchronous depending on requirements) with the SMSC. The message data is streamed across this link to the centre where it is processed and forwarded to the relevant mobile handset(s).

There are a number of advantages to this method. Firstly the third party, which is effectively a "middle man", is removed from the equation so the cost is reduced. There is also no restriction to one-way messaging using this method. When a connection is established between the project server and the SMSC, it is possible to build or edit distribution lists for different user groups. The most useful feature of this type of connection is two-way messaging. It is possible to bind to the SMSC using a range of mobile numbers and then retrieve all waiting messages for the bound set of numbers. So messages can be retrieved by the project server and forwarded to the user in the form of a web page. This web interface may then allow the user to process these messages as if they were emails. They can be stored permanently in a database and retrieved at any time through this interface. If all messages are stored on the project server database then there is no restriction based on SIM card size so any number of messages can be kept.

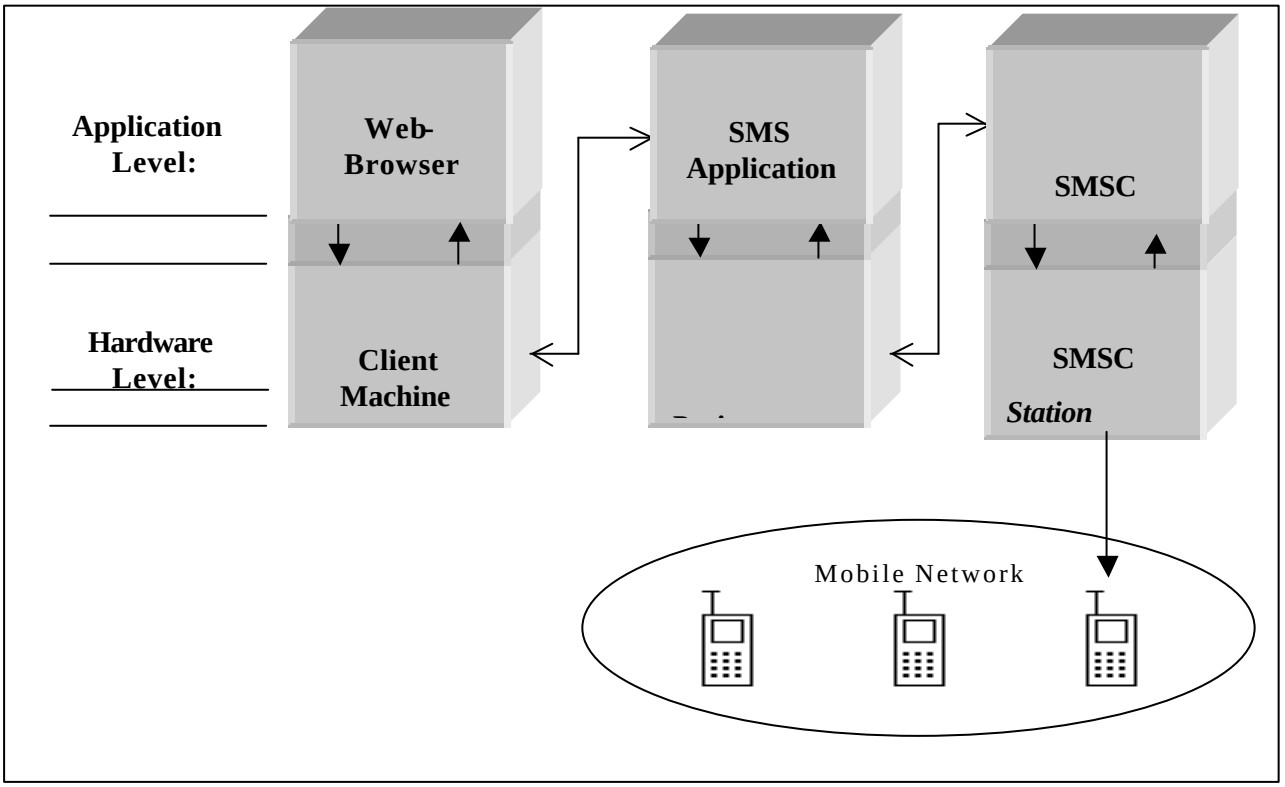


Figure 3: Diagram of the TCP/IP Method

Another advantage of this approach is paging functionality. Short Message Service and Number/Message Paging are based on the same basic technology. SMS is merely a functional superset of paging. Therefore it would be simple to incorporate a paging module to coincide with the SMS module. Figure 3 has bi-directional arrows connecting the client, project server and SMSC together. These show the two-way communication possible over the TCP/IP link as opposed to the one-way "message submission" method possible over HTTP alone.

Advantages of HTTP Method:

- Easy and very fast to implement.
- No responsibility for quality of service.

Advantages of TCP/IP method:

- Two-way messaging.
- Possible to retrieve messages on computer instead of mobile handset.
- Possible to store more messages on database than normally possible with mobile SIM cards.
- Possible to extend use of stored messages, e.g., forward messages on as email without re-writing.
- Extend functionality of SMS server application to also include the paging protocol.
- Two-way messaging via one web interface is the core idea of this project.

10. Preliminary Conclusions

The convergence of telephony, mobile services and the Internet strongly suggests that future UMS design should be based around IP. There is already widespread support for text based messaging in IP networks. The online email paradigm seems well suited not just to email bit to integrated text messaging. Research indicates that SMTP with MIME, IMAP4 and LDAP will be core protocols in our overall design. With well-established IP gateways to the Mobile Networks, integration of SMS with Email will be very achievable. Our design will primarily rely on the use of TCP/IP connections but either the HTTP or air interface connection could be used for reliability. The next stage of this research will examine the integration of voice-based messaging into our design and establish an overall design framework for a system that unifies both real and non real-time communications.

References:

1. Avison, David and Shah, Hanifa (1997) **The Information Systems Development Life Cycle: A First Course in Information Systems**, McGrawHill
2. Ford, Andrew, (2000), **Apache - Pocket Reference**, First Edition, O'Reilly
3. Laurie, Ben and Laurie, Peter, (1997), **Apache – The Definitive Guide**, First Edition, O'Reilly

4. Dubois, Paul, (2000), **MySQL**, New Riders
5. Crispin, M., (1994), **RFC-1733 - Distributed Electronic Mail Models in IMAP4**, Network Working Group
6. http://www.cae.wisc.edu/facstaff/imap/mail.html, Computer Aided Engineering Centre
7. http://www.imap.org/, The IMAP Connection
8. http://www.noctor.com/smsjdk.htm, Noctor Consulting

The Process And Problems Of Business Start-Ups

Natasha Evers
Institute of Technology Blanchardstown
Natasha.Evers@itb.ie

ABSTRACT

"… there is little known about the initial phases of the process (of entrepreneurship). The conception, birth and early development of new ventures are very much an uncharted territory". (Reynolds and White, 1997:1).

This paper sets out to examine the process and problems encountered by new business start-ups. A didactic overview, based on past and current literature in the field, identifies the most common theoretical frameworks frequenting the academic literature and assesses their contribution to explaining and understanding the Process and Problems of New Venture Creation.

The founding of a new organisation is not instantaneous and the process is a largely complex one. The nature of this process - which is characterized by spontaneity and uncertainty - makes it more difficult to pin down an exact theory. As Gartner (1985) points out, entrepreneurial firms are too diverse to permit generalization, and the process of starting up a new business has become a multidimensional phenomenon. The different approaches, suggested in literature, explaining the process of new venture creation, have attracted much academic controversy, given the lack of consistent empirical research on the process of new business creation. In this light, the author suggests that a more holistic understanding of the process may be gained through the integrated theoretical frameworks of new venture creation presented in the literature, which aim to capture the most important variables and characteristics of the new venture creation process.

The second part of the paper deals with the problems facing entrepreneurs in new venture creation. Many start-ups never reach establishment, and the majority close up within one year after they have become established. Embarking on a new business is one of adventure and challenge but it brings with it high risk and uncertainty. This paper does not seek to detail each and every industry-specific problem that start-ups experience, but aims to identify and examine the most common difficulties encountered by Start-Ups in the early stages of establishment, irrespective of sector or industry.

1. Objectives of paper

1.1 Objectives And Scope

This paper will examine the process and problems encountered by new business start-ups in the field of entrepreneurship. It aims to provide a didactic overview based on past and current literature in the field, identifying the most common theoretical frameworks frequenting the

academic literature and assessing their contribution to explaining and understanding the *Process and Problems of New Venture Creation*.

Entrepreneurship is one of the youngest paradigms in management science (Bygrave 1989b) and there is no general agreement on the defining concepts and variables explaining it. This too is reflective of the Start-Up process in the study of Entrepreneurship, where little agreement can be made on a common theoretical framework (Bhaves, 1995). In practice, the founding of a new organisation is not instantaneous and the process is a largely complex one. It evolves over time, as one must seek resources and compete in the marketplace. In much of the literature, this process of establishing the entrepreneurial start-up is characterized by both uncertainty, in terms of outcomes, success, failure, survival, lack of knowledge and understanding (Deakins & Whittam, 2000:116). Reynolds and White (1997:1) comment that " there is little known about the initial phases of the process (entrepreneurship). The conception, birth and early development of new ventures are very much *an uncharted territory*".

The different approaches, suggested in literature, explaining the process of new venture creation, have ignited much academic controversy. Moreover, there exists little consensus found across empirical studies for describing the process of new firms upon initiation. Despite the limitations in empirical evidence and diversity of academic opinion, insight can be gained by adopting important and empirically tested aspects of these different approaches and models to explain the start-up process. The most relevant aspects can be integrated into a theoretical framework to encapsulate the important stages and events encountered by start-up ventures. The scope of paper will be limited to the actual process of firm creation – from idea conception to establishment of the new organisation – and the problems encountered by firm and individuals during the start-up process.

Before launching into the theoretical approaches, the next section presents definitions and key terminology of the topic in question.

1.2 The Concept of "Process of Business Start-Up"

The stages leading up to the legal creation of the organisation, when it becomes an organisation or active legal business entity, is also referred to in biological terms - th*e journey from conception to birth*. Gartner (1985) has referred to this process of starting up as one which involves events before an organisation becomes an organisation, that is, organisation creation involves those factors that lead to and influence the process of starting a business.

Weich (1979) defined "New Venture Creation as the organizing of new organisations, …to organize is to assemble ongoing interdependent actions into sensible sequences that generate sensible outcomes".

A number of researchers have labeled this time period in an organisation's life as "Start-up"(Van De Ven, Angle & Poole 1989:Vesper, 1990), Preorganisation (Katz & Gartner, 1988; Hansen 1990), Organisaton in Vitro (Hansen & Wortman 1989); Prelaunch (McMullan & Long, 1990); Gestation (Reynolds& Miller, 1992;Whetten, 1987); Organisational Emergence (all cited in Gartner et al 1992: 373). These all refer to the same phenomenon.

Reynolds and Miller (1992) referred to the start-up process as a *biological* cycle in that the process can be described as a "gestation process" from conception to birth. There has been little study on the gestation of firms. The authors admit that it is very complex to identify when the idea of the firm has been conceived or when does the initial idea for a business come about. The answer is "we do not know". The process leading up to the birth of the firm is still largely a mystery and the actual duration of gestation has not been determined. Nevertheless, empirical evidence has shown that a common indicator for the birth (birth date) of the firm has been usually the date of its first sale as a sign that the firm is active participant in the economy (Reynolds & Miller, 1992). It is understanding the conception of the idea and the events leading up to the birth of the new business entity which has become the real challenge for academics.

The individual(s) which finds, manages and drives the creation of the new firm is commonly referred to as the *nascent entrepreneur*. Endemic to the process of business start up, the backgrounds and character traits of nascent entrepreneurs have been a common theme of research in understanding the start-up process (Entrepreneurial Trait approach will be looked at in the section 2.4). Reynolds (1994) suggested that nascent entrepreneurship should form part of the process and not just outcomes. He identified the principles of networking, resource mobilization, and entrepreneurial enactment should be inherent to the process of creating new business.

2. *MAIN THEORETCIAL APPROACHES TO NEW VENTURE FORMATION OR START-UP PROCESS*

The theoretical frameworks in literature have aimed to provide an understanding and explanation to the process of venture formation and factors influencing its creation. As noted earlier, diversity of opinion and little consistency in empirical evidence have prevented new

venture **creation process being underpinned to one paradigm.** As there appears no one best way of understanding this phenomenon, this section will identify the key models that have attempted to explore this area.

2.1 Some Macro Perspectives of New Firm Creation

2.1.1 Schumpterian Conceptualization of The Entrepreneurial Creation Process

Joseph Schumpeter's book titled " The Theory of Economic Development"(1912), was the first to refer to the creation of new firms as a factor of economic development. Schumpeter believed that the emergence of a new firm depended on the entrepreneur's identification of opportunities for combinations of production factors, which can result from technological change. The Schumpeterian model of new firm creation is illustrated in figure 1.

Schumpeter postulated that innovation is a central factor in new firm creation and the driving force behind industrial change. According to Schumpeter "A perennial gale of creative destruction" brings firm creation – destruction is the price of innovation leading to the emergence of new firms in economies. He proposed that if innovation can determine the speed of structural change in industry, then technological change acts as the "cue" for the leading on of new firms to take the stage.

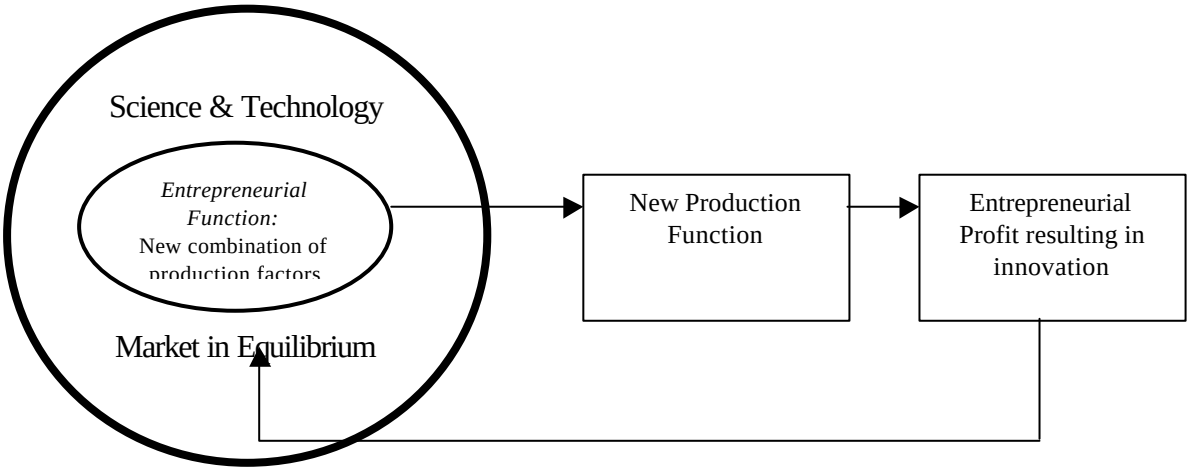


Figure 1: Schumpeterian Model of Entrepreneurial Function (Veciana, 1999)

Schumpeter's views may be useful for explaining the emergence of new firms in high growth sectors characterized by short product cycles such as Internet services, telecoms, electronics, games and entertainment software; where the rate of product innovation is very high. From a macro perspective, Schumpeter's economic theory may explain the creation of new start-ups in this these dynamic highly innovative industries here-mentioned.

2.1.2 Population Ecology Theory

Population ecology theory (Hannan & Freeman 1977) assumes that the external environment determines the entire existence of a new firm from the beginnings at birth, growth and death. It takes the population of the organisations as the unit of analysis and examines the external environment i.e., structural, economic and political conditions, that lead to the creation of new forms of organisations. Hannan & Freeman (1984) propose that organisational death rates should decline monotonically with age because organisational learning and inertia gradually increase with age. The emphasis is on the resources in society, not individual motives, nor decisions or behaviour as the driving force behind the creation of organisations. Hence, one could argue that view contradicts the classic notion of the entrepreneur who is regarded to hold the *locus of control and determine his or her own destinies*. Nevertheless, from what follows, the population ecology offers a valuable insight into understanding the pluralistic emergence of new firms in industries.

The population ecology approach to explain the birth of new firm is a macro perspective of the emergence of new organisations and tells very little about the process of starting-up at firm level. The process itself is beyond the individual and firm control (as already mentioned) and thus this theory gives no insight to understanding the process of venture creation at micro level. However, from a macro perspective it provides insight into the creation and cessation of new firms and why and how new organisations emerge in sectors, industries, communities and economies - an important area of study for public-policy makers. Furthermore, this stream of population ecological research has provided valuable knowledge into time-dependant patterns of organisational demography, particular for new firms (Van de Ven, 1992). Aldrich (1990) indicates, the ecological perspective stresses that new firm start-ups are highly dependant on the macro processes both within and between organisational populations. A body of population empirical evidence has demonstrated the consistency of this theory across a number of sectors. These include: Newspaper, Automobile, Brewing and Semi-conductor sectors (Veciana, 1999). The Population Ecology Theory and its supporting empirical evidence has stimulated scholars of entrepreneurship to examine more macro related questions regarding the factors that influence the rates of organisation births and deaths.

2.2 Literature Review of Approaches to New Venture Creation

A number of academics have presented frameworks for discerning the characteristics of the venture creation process. A summary of the key frameworks (Gartner, 1985; Gartner & Katz, 1988) is provided below and will be explored later in this section.

Gartner (1985) outlined a framework of four dimensions to be considered whilst studying new ventures: **1)** the individuals involved in the creation of the new start-up; **2)** the activities undertaken by those individuals during the new venture creation process; **3)** the organisational processes structure and strategy of the new venture; **4)** and the environment factors under which it must operate. Also in a series of stages, Van de Ven et al (1989) proposed that researchers must take account of **1)** how a business idea emerges over time, **2)** when and how different functional competencies are formed to develop and commercialize the first market offering, **3)** when and how these functional competencies are redeployed to create subsequent new line products believed to sustain and grow the business, finally **4)** how these efforts for business development are both influenced and constrained by organisation and industry factors (N.M. Carter et al, 1995: 153).

Karl Vesper (1990) contended that a new start-up has five components: 1*) technical know-how; 2) a product or service idea; 3) personal contacts; 4) physical resources; and 5) customer orders.* Vesper proposed a number of start-up sequences that vary among the five key ingredients (Timmons, 1980). Probably the most pioneering work was carried by J. Katz and W. Gartner (1988) who explored the organisation theory and entrepreneurial literature to identify a theoretical and empirically based framework for identifying the properties that would signal that an organisation is in the process of creation. In their framework, (which will be dealt with in section 2.6.2) the authors suggested four emergent properties that would indicate that an organisation is in the process of coming into existence: *intention to create an organisation, assembling resources to create and organisation, developing an organisational boundary, and exchanges of resources across the boundary* (e.g. Sales). In the last decade, integrated frameworks based on past models have emerged (Veciana 1988, Bhaves, 1995; Deakins & Whittam, 2000). These aim to provide a more comprehensive model to understanding the phenomenon and have attempted to encapsulate the key characteristics and variables describing the process of new enterprise formation in their proposed frameworks.

2.3 Early Approaches to New Venture Creation Process – Systematic Models

No single model or isolated sequence of events can apply to all start-ups during their process of creation. According to J. Timmons (1980), *Trial and Error* replaces the sequence of events that had traditionally been applied to describing the start-up process in literature. Equally, Gartner (1985) concluded that *entrepreneurial firms are too diverse to permit generalization*. However, in the 1970s, a systematic approach to understanding the process of start-ups was quite popular amongst academics. They proposed the firm being created would follow a sequence of mechanical steps before it could establish itself as a legal business entity. Flow charts were also common models outlining stages in the venture creation process. In his Article, titled (1980), "New Venture Creation: Models and Methodologies", J. Timmons undertook a review of the models on the venture creation process. As noted earlier, K. Vesper (1979) proposed five key ingredients for creating a business. Timmons equally contended that five key components were required to start a firm. There existed over 100 sequences to new venture creation and start-up process. Birley (1984) proposed eight events in the start-up process. These events were assumed to occur in the following order: 1*) owners decision to start a firm; 2) own quits job and becomes self-employed; 3) incorporation; 4) bank account established; 5) premises and equipment obtained; 6) first order received; 7) tax first paid; 8)first employees hired* (Reynolds & Miller, 1992).

No consensus existed amongst academics as to what was the correct systematic model. A common denominator of these process models was the individual as initiator of the business – the Entrepreneur. The numerous models outlining sequences and stages to new venture creation were theoretically based on assumptions, which gave very little insight into current practices at the time. This came as no surprise in light of the absence of empirical evidence to support them. However, these sequential models served as a basis for subsequent research.

2.4 Entrepreneurial Approach (Trait Approach)

The Founder / Nascent Entrepreneur of the new organisation is perceived as the key determinant of the firm creation in this approach. This is the classic approach to venture creation in entrepreneurial literature, which has mainly focused on the traits and behaviours of the founders with little or no attention paid to organisational and environmental factors to explaining the process of start-ups (Aldrich & Wiedenmayer, 1993).

This approach states that there exist linkages between individual traits and organisational creation (Van de Ven et al 1984). The individual is the unit of analysis in the organisational creation and innovation. This approach devotes attention to the background, characteristics, and psychological make up of the entrepreneur as key indicators of firm creation and its performance. Motives and personal attributes are considered to be strong influential factors in differentiating entrepreneurs with non-entrepreneurs. The concentration on entrepreneurial traits, such as character and hard work has been the dominant theme for explaining entrepreneurial achievement. However, this approach has lost its popularity amongst academics in entrepreneurship. Research has consistently found that personal traits, taken out of context, offers little understanding of this phenomenon (Aldrich, 2000). According to Gartner et al (1988) research on personal traits have reached an empirical *cul-de-sac*. Focusing on personal traits and character alone are no longer accepted for explaining the complex process of starting a business.

2.5 Human Capital /Knowledge Approach to Start-up Formation

Most organisation founders identify opportunities for creating new organisations from expertise, experience and education gained in previous work organisations (Aldrich 2000). Researchers have only begun to devote attention to these factors in the study of organisation creation. The nascent entrepreneur´s past experience, education and skills set can affect the formation of business ideas and the ability to start successful enterprises. This accumulation of experience and "know-how" is termed "*Human Capital*". The formulation of business ideas may be influenced by work experience, training and by recognition that a particular product or process could be done better. Education can play an important role in creating an inductive environment for idea formulation. Importance is also placed on "enterprise abilities including problem-solving, group work and idea generation". Timmons (1994:43) states "the notion that creativity can be learned or enhanced holds important implications for entrepreneurs who need to be creative in their thinking". Thus education can become an important conditioning experience. Creative thinking can be enhanced by the education system, which may affect the way opportunities are viewed later on in life (Deacons & Whittam, 2000).

According to H. Aldrich (2000), nascent entrepreneurs require several kinds of knowledge such as work experience, advice from experts, and copying existing organisation forms. This focus on human capital has been regaining importance as a key factor in understanding and explaining why and how start-ups emerge. An extension of this knowledge factor in the start-

up creation process is the *networking ability* of the entrepreneur to accumulate and leverage knowledge. Although it is worth noting that networks have direct linkages with human capital as described here, the role of Networks in business creation, given their importance, will treated separately in section 2.7.

2.6 Organisational Approach to New Venture Creation

The organisational approach focuses on the *process by which founders construct new organisation*. It posits that the creation of an organisation is not down to the individual founder or entrepreneur, but it is a collective, network building achievement that centres on the inception, diffusion and adoption of a set of ideas among a group of people who become sufficiently committed to these ideas to transform them into a social institution (Van De Ven et al, 1984: 95).

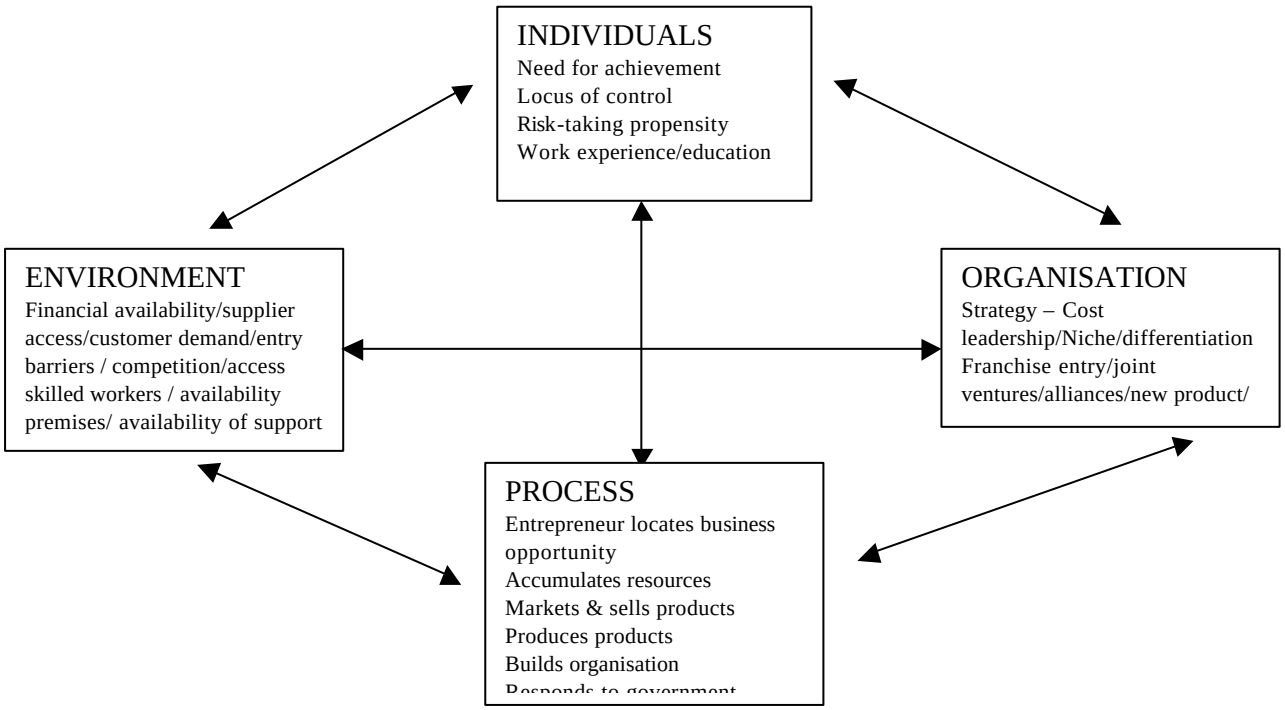
This view contends that the conditions under which an organisation is planned and the processes followed in its initial development have important consequences on its structure and performance later on in its business life cycle. The organisational approach to new venture formation has attracted much attention by scholars (Gartner, Katz, Vesper & Van de Ven are prominent writers in the field) and has become a widely accepted point of reference amongst the academic community for understanding the process of start-ups. This section focuses on two popular frameworks building on the organisational approach to new enterprise formation: W. Garter's (1985) 'Conceptual framework for describing New venture Creation' and second W. Gartner and J.Katz pioneering paradigm on 'Properties of Emerging Organisations' (1988).

2.6.1 Gartner's 'Conceptual Framework for Describing the Phenomenon of New Venture Creation' (1985).

Before his major work with Katz in 1988, Gartner (1985) proposed a conceptual framework for describing the phenomenon of new venture creation. Gartner contended that firms vary vastly in their characteristics as do the entrepreneurs who create them. He added the process should not be perceived as one-dimensional, carried out single-handedly by the entrepreneur. He argued that designing specific stages and fixed ingredients to form a new organisation which had been proposed by former scholars, and attaching a "type" of entrepreneur to each start-up was also too simplistic a process. Instead, Gartner argued that the process of new venture creation was a complex and multidimensional phenomenon.

In his 1985 framework, (see figure 2), he suggests a new enterprise is the result of four key variables – 1) Individuals - the person(s) involved in starting a new organisation; 2) Environment – the situation surrounding and influencing the emerging organisation; 3) New Venture Process – the actions undertaken by the individual(s), to start the venture; 4) Organisation – the kind of firm that is started. Each variable describes only one aspect of the process of venture creation and is interdependent on other variables in the framework. He adds that entrepreneurs and their firms do not represent a 'homogenous population' as previously assumed. Entrepreneurs and firms differ greatly in actions; choices; behaviour; environments they operate in and how they respond to internal and external situations. Gartner points to the importance of recognising this variation as a key characteristic in the process of new firm creation where it is unacceptable to "focus on some concept of the "average" entrepreneur and "typical" venture creation (Garner 1985, 697). A study conducted by Cooper ad Dunkelberg (1981) empirically backed up the logic of Gartner's argument on variation, revealed that entrepreneurs in certain industries can be very different from those in other industries. Similarly, Karl Vesper (1979) a famous scholar in the field suggested 11 types of entrepreneurs, also indicating early recognition of intrinsic variations in new venture creation processes. Gartner's framework has achieved much popularity for being able to highlight the diversity of entrepreneurs and firms and at the same time to encapsulate the complexity, intensity and diversity of this multifaceted phenomenon.

Figure 2: A Framework for describing New Venture Creation (Gartner 1985: 698) and some examples of variables in new venture creation



2.6.2 Katz & Gartner Framework (1988)

The most groundbreaking work in analysing organisation emergence has been the Katz & Gartner Framework (1988), identifying properties of organisations 'in creation' or 'emergence' (1988). The authors sought to identify when an organisation was in the process of starting up i.e in the 'preorganisation' period, since much research at the time was conducted in the 'organisation' period - after they were created. Based on B. McKelvey's definition of a organisation[1] , they suggested four key properties or characteristics that would determine whether an organisation was in the process of creation. These four properties were:

- Intentionality — intention to create an organisation
- Resources — assembling resources to create and organisation
- Boundary — developing an organisational boundary
- Exchange — exchanges of resources across the boundary

According to Gartner & Katz (1988), this framework can be used to identify the different ways in which a start-up process might occur based on these properties. The properties contain structural characteristics – *resources and boundary* - and process characteristics – *intentionality and exchange*. These properties are defined below.

Intentionality property refers to the intentions and goals of the founder(s) entrepreneurs and the goals of the various environmental sectors at the time of creation. These goals may span technology, equipment, capital, community etc. In the initial stages, the intentionality of an imminent start-up may overlap other agents' goals that are operating in their environment. As the start-up develops its goals it will become increasingly distinct from other entities in the environment and become itself a separate entity (Katz & Gartner, 1988). Intentionality would also require the would-be venture founder(s) to engage in the gathering of information to establish these goals with the aim of venture creation. The *Resource property* refers to the physical components – human and financial capital; property, raw materials - that combine to form an organisation. As resources accumulate, the need for funds increase. Delving into personal savings and borrowing from family, friends become apparent. As costs amount, external sources of financing are necessary, hence the entry of venture capitalists and

[1] McKelvey (1980) an organisation is "a myopically purposeful boundary-maintaining activity system containing one or more conditionally autonomous myopically purposeful subsystems having input-output resource ratios fostering survival in environments imposing particular constraints".(Gartner & Katz, 1988:430)

investors. *Boundary* is defined as the barrier conditions between the organisation and its environment (Katz &Kahn, 1978). The organisation has control over assets that fall within its boundary, however, it must establish a physical and legal basis for exchanging the resources it requires across its borders. When an organisation defines its boundary for example through incorporation, applying for tax number, establishment of physical offices, phone line etc., it creates its own identity and differentiates itself from other legal entities. Finally, *exchange* refers to cycles of transactions within the organisational boundary and outside with other economic agents (Katz & Gartner 1988:432). Exchange is necessary for the organisation to operate and must be conducted efficiently i.e. selling goods for no less than the cost of producing them. The other three properties must be in place before exchange processes can occur. These four properties of emerging organisation are necessary to make the transition to an 'organisation'.

The authors see these properties as useful tools for researchers to build models for analysing potential sources of new ventures in a way that allows the identification of organisations the process of creation (Katz & Gartner, 1988). Moreover, the ability to recognise organisations early in creation should prove beneficial for determining the success and failure of different strategies adopted in start-ups.

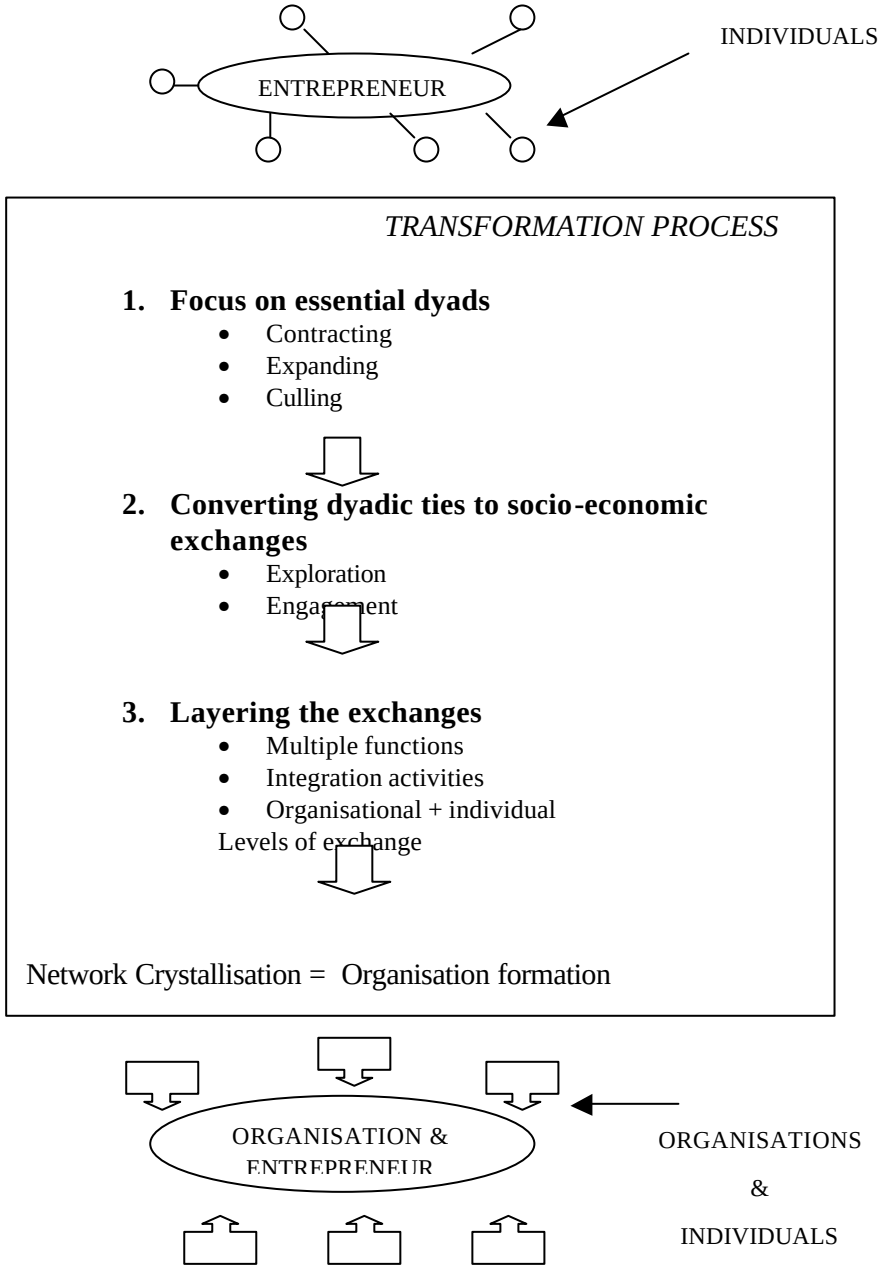
2.7 Network Approach (Social-economic model)

The role of social networks has become quite a fashionable approach to new venture creation. There have been many contributions to explaining networks as a factor in new firm formation. Nascent entrepreneurs´ personal networks are the set of persons to whom they are directly linked - impact their social and economic resources and support. Founders draw upon their network ties and build new ones as a means to obtain knowledge and resources for their emerging organisations. Aldrich (2000: 235) argues that nascent entrepreneurs "who occupy impoverished social locations may find themselves cut off from emerging opportunities and critical resources".

Larson and Starr (1993) propose a Network model (see figure 3) of organisation formation embodying socio-economic characteristics. Their model is a stage model that describes the processes by which the essential relationships between the entrepreneur and resource providers evolve to create an organisation. The model builds upon the theoretical and empirical research of network analysis. Although the network approach captures aspects of previous models (Katz & Gartner 1988) by linking the entrepreneur and the environment in which they

operate, it emphasizes the exchange processes between actors and units and recognises the social and economic aspects of these exchange relationships (Larson & Starr, 1993:6). The author's model is illustrated in figure 3 and outlines three stages that transform a preorganisation into a new organisation.

Figure 3: Network Model Of Organisation Formation (Larson & Starr, 1993:7)



According to Larson & Starr (1993), the outcome of the staged model of networking is the crystallization of an individual/organisational network made up of a critical mass of dyads that

establish the new organisation as a viable entity. The organisation has been formed mainly through collective activity between the various actors and units (network ties). This results in a stable, commitment of relationships between the entrepreneur and their resource providers. The network approach has been an accepted perspective for explaining a business start-up. It views the entrepreneur as a whole individual, a socio-economic actor with personal and economic interests. Their network model analyzes the role of economic relationships set in a social environment. This model is a socio-economic model emphasising people in relationships rather than pure economic units in transactions. It places a dual emphasis on social and economic dimensions of exchange.

The authors of the model see it as complementing the Katz and Gartner Framework (1988). They (Larson & Starr, 1993) consider the proposals of four organisational properties to "fit comfortably" with their network model. For instance, the structural properties are founding the pattern of mobilized resources. The boundary is defined by the extent of business relationships. The activities and intentionality of the founder and his/her network ties combined with the actual exchange processes described to constitute the process properties (Larson & Starr, 1993:12).

To exclusively rely on the network model to explain new venture formation would be a too simplified approach as it undermines the importance of the business idea, strategy, the industry and the abilities and skills of the entrepreneur themselves (J.M. Veciana, 1999). There has been empirical research (Birley, 1985, Aldrich et al 1986,) that could not confirm the role of networks as a key ingredient in the formation of new start-ups. Birley concludes "information on… the role of networks in connection with new venture creation is still scarce and anecdotal" (1985:85). Despite these findings, the amount of empirical research is too limited to be conclusive. Moreover, the study of the role of networks in the new venture creation process is still in its infancy, and requires more research. However, one cannot deny that co-operation and business relationships have always made sense for new businesses since Commerce began.

2.8 Integrated frameworks on New Enterprise Formation

The models presented above offer relatively broad categories and generalized variables which reveal no nuances in particular areas of the start-up process. However, they do capture important aspects of new venture creation, which numerous authors have integrated, based on

empirical evidence, in an attempt to present a generic theoretical framework. This section provides a sample of selected frameworks, chosen by the author, that have managed to integrate the key concepts and events of the process of venture formation. These models captures elements of each approach -Entrepreneurial, Knowledge, Network and Organisational – and presents them in a comprehensive and unified framework for explaining the process of new venture creation.

J. M. Veciana (1988) outlines four phases of venture creation process with an estimated timeframe as illustrated in figure 4. The author presents each of the most relevant variables occurring in each phase. The variable presented is one most likely to impact that particular phase on the process of venture creation. The five-year period of establishing a new venture may be a realistic one, in light of the activities the entrepreneur must undertake. Further comments on the author's interpretation of the model is restricted by the fact that the commentary is in *Spanish*.

Figure 4: Process Of New Enterprise Formation

		1-2 YEARS	2-5 YEARS
GESTATION	**CREATION**	**LAUNCHING**	**CONSOLIDATION**
Childhood Antecedents Incubators Critical Event/Role Determination Decision to create a firm	Search for & identify opportunity Elaboration & configuration of the entrepreneurial project Network creation Opportunity Evaluation Preparation of a Business Plan	Team Building Purchase & organisation of the production factors Product/Service development Search for finance Launching of product/service	Getting through the knothole Getting rid of partners At last "everything under control"

Source: Veciana, J.M. (1988): The Entrepreneur and the Process of Enterprise Formation, in "Revista Economica De Catalunya", Num.8,May-August.

Studies of entrepreneurship aim to generate generalised conclusions about variables relevant to all firms. Yet on the other hand, each enterprise is unique and conceived by individual and personal means, with varied circumstances facing entrepreneurs throughout start-up process and when the firm is established.

P. Bhave's paper (1994) presents a *Process Model of Entrepreneurial Venture Creation.* The author aims to provide a "well-grounded-theoretical, integrative process model of entrepreneurial firm creation by linking conceptual categories and sub-processes in the firm creation process based on qualitative research" (Bhaves, 1994:225). The author states that this process model intends to provide an integrated framework to bring cohesion to the vast body of literature. This model is important as it extends its concern to sub-processes of venture creation, which have been largely ignored in literature.

A final and a most recent paradigm developed by Deakins and Whittam (2000) is illustrated in figure 5. The authors suggest that the business start-up process can be broken down into a number of stages:

- Formation of the idea
- Opportunity recognition
- Pre-start planning and preparation including pilot testing
- Entry into entrepreneurship launch
- Post Entry development

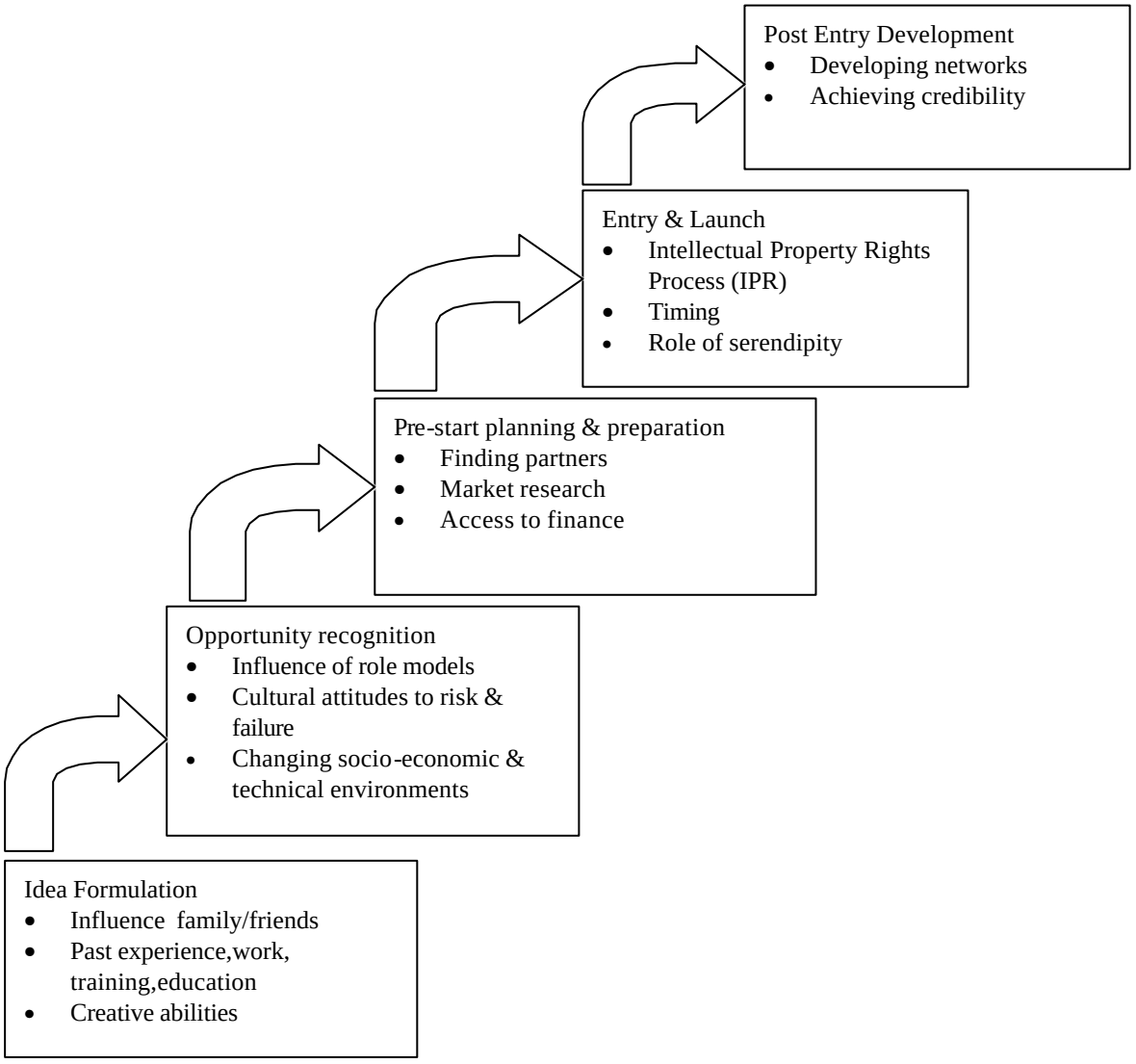
Each of these stages will consist of a number of positive and negative factors that impact the process of starting up. The authors also point out that the number of factors affecting each stage is not exhaustive, a host of other influences exist including the cognitive behaviour of the individual entrepreneur, such as tenacity and ability to over come obstacles to creating a new business (Deakins & Whittam, 2000).

2.9 Empirical Findings

Subsequent empirical explorations, (Reynolds & Miller 1992; Reynolds & White 1993; Reynolds 1994) confirmed the conclusions of the Katz and Gartner (1988) framework that no one pattern or sequence of events is common to all emerging organisations. Moreover, the sign of the exchange process of the "first sale" is a conceptual event in new venture creation (Block & MacMillian, 1985). First sale has been used as a measure of successfully establishing a business based on Katz & Gartner's (1988) properties of emerging organisations framework as described above. Reynolds and Miller's (1992) study of new firm gestation indicators concludes, "Date of first sale appears to be a suitable indicator of "Birth" if only one event is to be used (p. 406).

An interesting study exploring 71 US based nascent entrepreneurs in new venture organisation was conducted by Carter, Gartner and Reynolds (1996). This study was based on primary and secondary data. According to their findings, what was most common as a first stage in the start up process was the *personal commitment* by individuals engaged in the new venture (five out of six firms), some emerging organisations (two in five) reported the first event as having sales, whereas others began recruiting or seeking financial support (one in four). The most common final events in the process of new business creation was recruiting employees and making sales (half of new ventures), financial support (two in five), and a huge personal commitment to the venture (one in four). In addition, Reynolds et al (1994) discovered that the average time a firm was in the process of creation was one year. In contrast, Van de Ven et al (1990), in a study of hi-technology firms, found that the average time for entrepreneurs to create the business from inception to birth was four years. (N.M. Carter et al, 1996: 154).

Figure 5: Business Creation And The Start-Up Process: A Suggested Paradigm



Source: Deakins & Whittam, 2000:121

An important field of research that has emerged in recent times is the role of cognitive psychology in new venture creation (Aldrich, 2000, Gatewood et al, 1995; Carter et al; 1996). Gatewood, Shaver and Gartner (1995) carried out a longitudinal study of individual level factors (cognitions and actions of the entrepreneur) influencing the process of starting a new business. The primary focus was to determine what appropriate measures could be used to identify cognitive factors, which might influence an individual' persistence in entrepreneurial activities despite the uncertain chances of start-up success (Gatewood et al, 1995). The researchers concluded that by doing longitudinal research design, stronger claims could be made between the relationship between individual attributes and subsequent success in starting a venture.

A second study, conducted by Carter, Gartner and Reynolds study (1996), revealed that cognitive factors played an important influence on the process of starting a business. The study suggests that the behaviours of nascent entrepreneurs who have successfully started a new venture can be identified and differentiated from the behaviours of the nascent entrepreneurs who failed. However, the precise type of behaviours appropriate for new venture conditions were not identified and would require being studied in future research.

3. COMMON PROBLEMS FACING NEW BUSINESS START-UPS

Creating a new business is fraught with difficulty and failure (Reynolds & Miller 1992;Van De Ven 1992b). Many start-ups never reach establishment, and the majority close up within one year after they have become established. Embarking on a new business is one of adventure and challenge but it brings with it high risk and uncertainty. Although some start-ups survive and become highly profitable, empirical evidence has shown that there exist key problems, which are common to all new start-ups regardless of level of innovation in their new product, the sources of finance, business experience, knowledge, and networks ties of the entrepreneur. Raising capital, establishing reputation, securing resourced providers, premises constraints and high labour costs have been recurrent problems stated in the literature and also in empirical evidence (Storey, 1985). This section does not seek to detail each and every industry-specific problem that start-ups experience, but aims to identify and examine the most common difficulties encountered by Start-Ups in the early stages of establishment, irrespective of sector or industry.

Many entrepreneurs, who possess the initiative and incentive to start their own business, often lack business experience in the industry they wish to compete in. However, some successful businesses were started by inexperienced founders, for example Bill Gates and Michael Dell were college dropouts. Steve Wozniak, founder of Apple Computers, "was an undistinguished engineer at Hewlett-Packard",(Bhide, 2000:36). As well as lack of experience, the nascent entrepreneur tends to have limited knowledge of the industry they enter. Most start-ups lack innovative ideas or assets that could differentiate them from their competitors. In Bhide´s survey of the 1989 *Inc*.500 list, a compilation of the fastest growing privately companies in the United States, he found that only 10% of these businesses offered novel product or services when start-up, with the majority of firms offering nothing original or new to the market.

Bhide (2000) conducted a further survey of all the Inc. 500 founders, between 1982 and 1989. He discovered that 12% of the founders attributed the success of their companies to "an unusual or extraordinary idea"; 88% reported their success was mainly due to the "exceptional execution if the idea", (Bhide, 2000:32). However, most new businesses which pursue an unnovel idea turns out to be unprofitable, and equally encounters more problems in their start-up phase. The widespread lack of innovative ideas, often accompanied by limited experience and knowledge can create huge barriers in raising capital.

Obtaining external financing is one of the key factors if the not the most in preventing start-ups from growth and development. The economics of information suggests that asymmetric information plays an important role when an entrepreneur seeks external financing for their new venture. In theory, when conditions of uncertainty combine with asymmetric information (where investors and borrowers have different sets of information), for the funders there are problems of selection (choosing profitable ventures) and moral dilemma (what will entrepreneurs do with this invested capital).

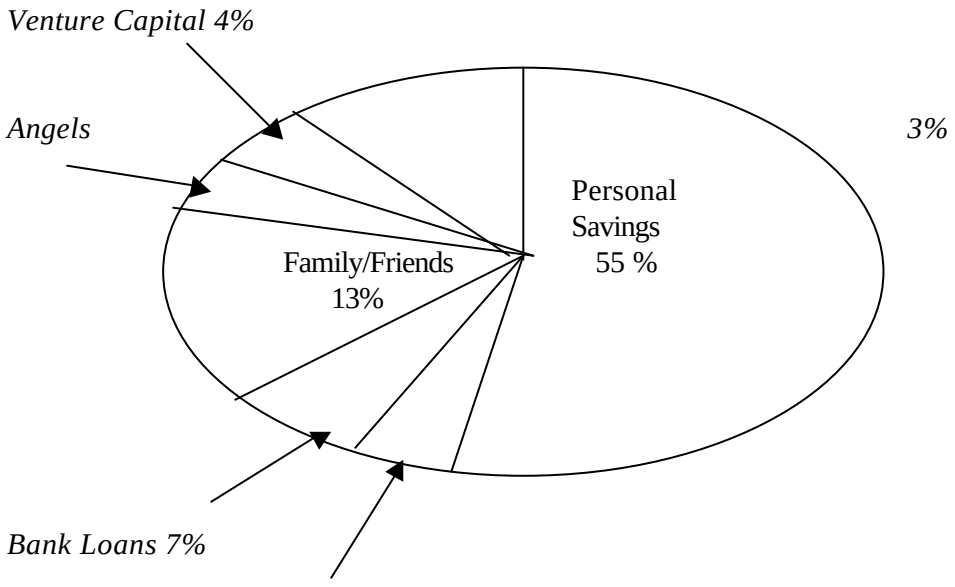
Most Entrepreneurs use their own personal finance as seed capital as venture capitalists and private investors require a strong highly credible business venture to ensure a return on their investments and recuperate their costs. Start-ups face disadvantages as they have a non-trading track record and may not have sufficient information to make risk assessments (Deakins & Whittam, 2000). The pie chart one illustrates that of the 1996 Inc. 500 companies, venture capital made up only 4% of start-up funds, with over half raised from personal savings. This indicates the enormous difficulties for even potentially properous start-ups to raise finance.

The inability to raise sufficient capital can lead to a negative 'knock-on' effect throughout the start-up process such as constraining expansion, problems with attracting clients and building alliances, and establishing credibility. Bannock (1981) commented that raising external finance is an inevitable problem of a business start-up. Banks dealing with a myriad of start-ups see them as administration and financial burdens than with large established firms. Moreover, start-ups are penalised immediately for having no commercial or financial history – an undermining factor to its credibility as a business entity.

Low capital resources can prevent the start from acquiring adequate premises particularly if its demand for orders requires larger premises. This *premises constraint* can restrict growth and may cause the start- up to refuse these orders and eventually lead to closure and eventually renders missed entrepreneurial opportunity.

PRIMARY SOURCE OF INITIAL FUNDING
PERCENTAGE OF 1996 Inc. 500 companies

Pie Chart 1



Source: O. Bhide, 2000.

The low endowments of knowledge; unique product offering can not only render it impossible for entrepreneurs to raise significant capital but also to secure resource providers, particularly potential customers and clients. Securing relations with resource providers such as customers, suppliers, employees etc. represents a critical problem for start-ups (Bhide, 2000). From the resource provider perspective, there exists a greater perceived risk dealing with a new firm

than with an established company with a track record. Establishing a market presence and securing customer orders is particular difficult for start-ups.

A. Bhide (2000) treats the problem of *securing customers* as resource providers for start-ups under two headings: first, Rational Calculus of Resource Providers – that is the choices that traditional economic models assume people typically make to maximize their utilities - present difficulties for nascent entrepreneurs. Second, he terms "Behavioural Factors" – he refers to this as deviations from rational decision-making due to cognitive prejudices (Bhide, 2000:70).

According to Bhide (2000), a main concern of resource providers is the level of switching costs. Customers need to assess the chances of survival of the new firm before it makes a commitment of time and money and incurs potential costs by switching over. The level of uncertainty increases even more on the part of the resource provider, should the start-up be undercapitalized. The fact that the entrepreneur failed to raise capital as well as a "zero" track record gives a negative signal to potential clients. The latter may also believe there is a good reason to be skeptical about doing business with the start-up if the investors rejected it as a potential investment. In other words, the start-up is perceived as a "non-credible" business entity. Thus lack of external financing can itself raise negative perceptions of start-up amongst resource providers. The rationale behind the resource providers' decision is to "let someone else go first". This approach leaves little chance for start-ups to survive and as Bhide adds "Luck " plays an extremely important role in the success of new ventures (2000).

Cognitive Biases of the resource providers can be good enough reason to avoid start-ups. Past experiences with failed start-ups, general gossip about start-ups regularly "going bust" within a few months of setting up can lead the resource provider to automatically refuse to do business with start-ups.

Based on empirical studies conducted in the UK and the USA, a major problem experienced by start-ups was *establishing a reputation* when there is shortage of demand in the marketplace (D.Storey, 1985). These studies also revealed that factors in the macro-environment such as interest rates, inflation and labour costs raised significant difficulties for start-ups (D.Storey, 1985). A further inhibiting factor to the process of starting up is acquiring legal recognition of a business. Government regulations can be quite stringent in developing countries where registering a new company is a time-consuming and costly process.

Finding solutions to the above problems and even avoiding them is difficult, and in the real world the process of starting a new business will never be problem-free. However start-ups in the event of encountering such problems may be able to mitigate the effects to a certain extent by being adaptive, flexible and alert to opportunity and threats in the market place. Establishing contacts through networks are equally important but *luck* also has a part in the process.

4. CONCLUSIONS

At macro level, the views of Schumpeter and the Population Ecologists have made a valuable contribution to explaining the emergence of new firms. Their theories are useful in explaining why and how new organisational forms come about and with so much variation. However at micro level researchers are stilling grappling with understanding the complexity of the entrepreneurial process of new firms. The nature of this process, which is deeply characterized by spontaneity and uncertainty, makes it more difficult to pin down an exact theory. As Gartner (1985) pointed out, *entrepreneurial firms are too diverse to permit generalization,* and the process of starting up a new business has become a multidimensional phenomenon. As indicated earlier, there has been little agreement on dimensions and variables characterizing it. The processes and birth of firms are not well understood (Reynolds & Miller, 1992, Low & Macmillan, 1988). Equally, there exist few empirical studies exploring and identifying conceptual categories and sub process of venture creation (Bhaves 1994).

Despite these research gaps, some common characteristics of start-ups have emerged in literature. The initial models, describing start-up sequences, served as a starting point and stimulated further study on the process of new venture creation. Gartner points to the importance of recognising this *variation* as a key characteristic in the process of new firm creation, adding that entrepreneur and their firms do not represent a 'homogenous population' as previously assumed. Entrepreneurs and firms differ greatly in actions; choices; behaviour; environments they operate in and how they respond to internal and external situations.

This observation on the "variation" concept in essence is *truistic*, there exist many variables impacting the process of start-ups which brings about much diversity and variation in today's business environment. Gartner in his work with Katz (1988) made another important development by using the four properties to identify when an organisation is in creation. The main achievement of their work drove home the point that organisation emergence is not a

linear step-by step process (Aldrich, 2000). Other key developments have emerged in recent times these include the human capital, network approach and the role of cognitive factors in the entrepreneurial process of new venture creation. These approaches have highlighted important aspects for explaining business start-ups however more empirical research is required.

From my analysis, there exists no single best approach or model that best describes and explains the new venture creation process, and which encompasses all its aspects and characteristics that have been mentioned in individual approaches. Integrated frameworks have been suggested as an attempt to solve this problem. Authors of such approaches as mentioned above (Veciana, Bhaves and Deakins and Whittam) seek to offer a more comprehensive holistic approach by encapsulating all the important variables and characteristics of preceding models on the venture creation process. Despite these attempts to offer an all encompassing framework, these variables are loosely defined, where more specific factors are needed. The weaknesses of the theoretical frameworks, presented in this paper, is that their authors have wanted to be "everything to everyone" but with little success. On the other hand, how can there be one generic model that can be applied to all start-ups in all sectors of the economy and to all nascent entrepreneurs? This proposal is not viable as firms and their founder(s) are too diverse, that there exists too little uniformity in the business environment to develop such generialised model or framework. What may be more productive for future research is to develop more specific models for new start-ups and their founders in *particular sectors* of the economy, this, I believe, would be a more realistic and viable path for research to take.

5. BIBLIOGRAPHY

Aldrich, H. (2000): Organisations Evolving. Cambridge: Sage Publications

Bhaves, M.P. (1994): " A process model of entrepreneurial venture creation", *Journal of Business Venturing* 9: 223-242

Bhide, A. (2000): *The Origin and Evolution of New Businesses,* Oxford University Press Inc.

Bygrave, W.D. (1993): "Theory Building in the Entrepreneurship Paradigm", *Journal of Business Venturing,* 8.

Carter, N.M., Gartner, W.B., & Reynolds, P.,(1996): "Exploring Start-Up Event Sequences", *Journal of Business Venturing* 11: 156-166.

Deakins, D. & Whittam, G.(2000): Business Start-Up: theory, practice and policy. In *Enterprise and Small Business Principles, Practice and Policy,* eds. S.Carter & D. Jones-Evans, 115-131. UK: Prentice-Hall.

Gartner, W.B. (1985): "A conceptual framework for describing the phenomenon of new

venture creation", *Academy of Management Review* 10(4): 694-706.

Gatewood, E.J., Shaver, K.G., Gartner, W.B. (1995): "A longitudinal Study of cognitive factors influencing start-up behaviours and success at venture creation", *Journal of Business Venturing* 10(5): 371-391.

Hannan J. &. B. Freeman (1984): Organisational Ecology, Cambridge MA: Harvard University Press. Cited in Van De Ven, A.H. (1992): Longitudinal methods for studying the process of entrepreneurship. In *The State of the Art of Entrepreneurship* eds. D.Sexton & J. Kasarda, 191-242. USA: PWS-Kewt publishing company.

Katz, J., & Gartner, W.B.(1988): "Properties of Emerging Organisation", *Academy of Management Review* 13: 429-441.

Larson, A. & Starr, J.A., (1993): "A network model of organisation formation", Entrepreneurship Theory and Practice 17(2): 5-16.

McKelvey, B. (1980): Organisational Systematics. Berkeley: University of California Press, cited in Reynolds, P. and White, S. (1993): "Wisconsin entrepreneurial climate study.

Milwaukee, WI: Marquette University Centre for the Study of Entrepreneurship. Final Report to Wisconsin Housing and Economic Development Authority.cited in Gatewood, E.J., Shaver, K.G., Gartner, W.B. (1995): "A longitudinal Study of cognitive factors influencing start-up behaviours and success at venture creation", *Journal of Business Venturing* 10(5): 371-391.

Reynolds, P. (1994): "Reducing barriers to understanding new firm gestation: Prevalence and success of nascent entrepreneurs. Paper presented at the Academy of Management Meetings, Dallas, Texas (August) cited in Gatewood, E.J., Shaver, K.G., Gartner, W.B. (1995): "A longitudinal Study of cognitive factors influencing start-up behaviours and success at venture creation", *Journal of Business Venturing* 10(5): 371-391.

Reynolds, P. & Miller, B. (1992): "New Firm Gestation: Conception, Birth and implications for research", *Journal of Business Venturing* 7: 405-417

MacMillian, J. C. & Long, W. A.(1990): "Developing New Ventures", San Diego, CA: Harcourt Brace Jovanovich, cited in in Gatewood, E.J., Shaver, K.G., Gartner, W.B. (1995): "A longitudinal Study of cognitive factors influencing start-up behaviours and success at venture creation", *Journal of Business Venturing* 10(5): 371-391.

Romanelli, E. & Bird Schoonhoven, C.(2000): The local origins of new firms. In The *Entrepreneurship Dynamic: Origins of Entrepreneurship and the Evolution of Industries*, eds. C. Bird Schoonhoven & E. Romanelli, 40-67. US: Prentice Hall.

Schumpeter, J.(1996): "The Theory of Economic Development", Transaction Publisher, New Brunswick Press, cited in Veciana, J.M. (1995): "Entrepreneurship as a Scientific Research Programme", Working Paper: European Doctoral Programme in Entrepreneurship and Small Business Management. UAB.

Storey, D. (1985): "The problems facing new firms", *Journal of Management Studies* 22(3): 327-345.

Timmons, J.A. (1980): "New Venture Creation: models and methodologies" in the Encyclopedia of Entrepreneurship.

Van De Ven, A.H.(1992): Longitudinal methods for studying the process of entrepreneurship. In *The State of the Art of Entrepreneurship* eds. D.Sexton & J. Kasarda, 191-242. USA: PWS-Kewt publishing company.

Van De Ven, A.H., Hudson, R. & Schroeder, D.M.(1984): "Designing New Business Start-ups: entrepreneurial,organisational and ecological considerations", *Journal of Management* 10:87-107.

Vanderwerf, P.A. (1993): "A model of venture creation in new industries", *Entrepreneurship Theory & Practice* 17(2): 39-48.

Veciana, J. M. (1988): "Empresari I process de creacio d'empreses", Revista Economica de

Catalunya, num.8. May-August.

Veciana, J. M. (1995): "Entrepreneurship as a Scientific Research Programme", Working Paper: European Doctoral Programme in Entrepreneurship and Small Business Management. UAB (and Revista Europea de Direccion y Economia de la Empresa), 1999, Vol. 9, No. 3.

Vesper, K. H. (1990): *New Venture Strategies*, 2nd ed. Englewood Cliffs, NJ: Prentice Hall.

Weick, K.E. (1979): The Social Psychology of Organising (2nd ed). Reading, MA: Addison-Wesley, cited in Gartner, W.B. (1985): "A conceptual framework for describing the phenomenon of new venture creation", *Academy of Management Review* 10(4): 694-706.

Whetten, D.A. (1987): "Organisational Growth and Decline Processes", *Annual Review of Sociology* 13: 335-358, cited in Gatewood, E.J., Shaver, K.G., Gartner, W.B. (1995): "A longitudinal Study of cognitive factors influencing start-up behaviours and success at venture creation", *Journal of Business Venturing* 10(5): 371-391.

Experiences gained from the deployment of an E-Learning "Java Arrays" Prototype for novice Java programmers in the Institute of Technology Tallaght, 2002/2003

Eamonn Hyland (Computing Lecturer, IT Tallaght)
Dean Fennell (Postgraduate Student, IT Tallaght)
Department of Computing
Institute of Technology Tallaght,
Dublin 24, Ireland.
eamonn.hyland@it-tallaght.ie
dean.fennell@openet-telecom.com

Abstract

This paper describes recent experiences gained from the deployment of an E-learning "Java Arrays" prototype for novice first year students within the Computing department at the Institute of Technology, Tallaght. It attempts to determine what contributions, if any, the E-Learning prototype made to novice students learning to declare, create, initialise and manipulate one-dimensional and two-dimensional arrays within the Java Programming language. Tentative conclusions from this process are presented and a charter for progressing this research further is outlined.

1. Introduction

This research paper is the second paper in a broader "Java and E-Learning" based research study that is being conducted in IT Tallaght since the academic year 2000/2001. The first paper ("Initial experiences gained and initiatives employed in the teaching of Java programming in the Institute of Technology Tallaght", Eamonn Hyland, Gary Clynch, PPPJ Conference 2002, TCD) sought to evaluate, among other items, the student's perceptions of our Java Syllabi and they were asked to rate topics they had covered in terms of levels of learning difficulty. The primary result of that study was that "Arrays" emerged as the most difficult concept that our novice student's had covered in terms of learning difficulties and levels of understanding. This outcome formed the basis and rationale for the development and deployment of an E-Learning "Java Arrays" prototype in the department to be utilised by the next group of novice students in the coming academic year. The E-learning prototype was developed and was made available to students in a strictly controlled fashion during November 2002. Data was gathered using a series of detailed and phased knowledge based questionnaires, which were presented to the students at regular intervals while they were studying the Arrays section of the Java software development stream. The overall aim of this research was to measure the effectiveness of the prototype in terms of its contribution, positive or negative, to the students learning experience.

The structure of this paper is as follows. Section 2 gives a brief description of the environment in which the study was undertaken. Section 3 describes in more detail the rational and design of the study. Section 4 describes the E-learning prototype as developed and deployed throughout the study. Section 5 provides the results of the study and presents an analysis of these results. Section 6 provides the conclusions reached as a result of this study.

2. Introduction to the study environment

The study was performed within the Department of Computing at I.T Tallaght. The student groups surveyed were completing the first year of a 2-4 year programme, which can lead after year 4 to a B.Sc. in Computing (Information Technology). There were approximately 180 students in first year when the study took place, divided into 10 classes for software development. The class sizes are deliberately kept under 20 students to facilitate timetabling of these classes into our 20 seater special purpose teaching laboratories (interactive whiteboards, teacher's console with synchronisation and presentation software to control the student's machines similar to an advanced training environment), outlined in detail in the initial paper. The software development courses are delivered using a combination of interactive lectures, tutorials, and laboratory sessions and are assessed with practical in lab assessments where students design and code their programs. The assessment phase tests the student's ability to apply the new skills they have learned solving IS programming problems.

3. Study rationale and Design

As stated in the introduction, this research was a natural follow on study based on the outcomes of the first paper (PPPJ 02, TCD). That paper had presented the results of an investigation into the success of various pedagogical initiatives employed teaching programming, and also questioned the student's perceptions of the Java syllabi, within the Computing Department at I.T Tallaght. To determine the student's perceptions of the syllabi topics, the researchers had conducted a qualitative based research survey. The study identified and weighted concepts in Java which students had difficulty understanding. Of the students surveyed, over a quarter (28%) identified Arrays as the most difficult topic. This main finding was considered significant in the context that these students were now in second year and had covered advanced topics such as classes, inheritance, encapsulation, but they still had difficulties with "Arrays", one year after covering it on the first year syllabus. This provided a basis and a rationale for designing and implementing an E-learning prototype exclusively

designed to help students learn and understand and apply arrays. The prototype would then be assessed to determine any identified contribution to addressing this identified learning difficulty with Arrays in Java. If the outcome of this study is encouraging or at least not in any way detrimental to learning, the prototype can then be further integrated into the Computing Department's software development teaching model.

Two student groups were chosen to partake in the study, one would use the prototype and one would not. The same lecturer was chosen to teach both of these groups separately to eliminate any bias with different lecturing styles. The prototype was installed in a Computing Department teaching lab on all students PC's in a controlled fashion. The prototype was made available to one set of students as a supplementary learning tool to the traditional methods while the other group was taught "arrays" using only the traditional methods already in place in the college, course notes, laboratory sessions and tutorials. The topic of arrays in Java, as taught within I.T Tallaght, covers 3 main sections 1) array initialisation and declaration, 2) array manipulation and 3) two-dimensional arrays. For the group using the prototype, they were required to write Java code statements or programs within the simulated environment to create and manipulate arrays. The group not using the prototype could use the Sun Forte IDE environment to do this as normal. Questionnaires were deemed the most suitable means of capturing data to meet the objectives of the study. Detailed questionnaires were designed based on a clear set of objectives to accurately capture how well students in both test groups understood the concepts in the 3 main sections. The questionnaires consisted of multiple-choice questions, true or false questions, as well as open ended questions such as "Write code to…", "Describe what is …", and "Explain this concept". An extra questionnaire was also designed which was distributed at the end of the study. This questionnaire was only given to students who had access to the E-Learning prototype. It was more qualitative in nature and it focused specifically on the prototype itself and how effective the students *felt* that the prototype was. The questionnaires were handed out at the end of classes in which a particular section had been completed. Students were allocated as much time as they needed to complete the questionnaires.

4. E-Learning prototype design and implementation

Various development environments were considered including implementing the prototype as a Java applet, using PERL, C++ based CGI scripts, HTML, Macromedia Flash, Java, TopClass and Blackboard. Flash was deemed to be a viable development environment due to its visual

strengths and its ability to deliver a fast web based application. In terms of distribution, it would require merely a plug in the target PC's, and in the Computing Department at I.T Tallaght it is already installed as part of our generic download. Flash uses a scripting language, ActionScript, which facilitates ease of implementation and debugging facilities. It provides powerful control of a rich Multimedia interface, which can incorporate and control graphics, text and sound. Using this development environment meant that the prototype could be converted into an executable form for use on a student's PC at home. Also as long as a host machine had the Flash plug-in installed the E-Learning prototype could be used remotely or streamed over the Internet from a remote server.

The prototype was designed with a strong visual GUI interface, to be simple and intuitive to use. In development terms, most of the effort was spent developing a Java Parser Engine. This Engine works behind the GUI to interpret Java individual code statements or indeed Java programs entered by the student. It can then represent arrays visually "on the fly" that were declared by the student's Java code. The Parser Engine consists of a basic lexical analyser/syntax checker that recognises a pre-determined set of Java statements and operators. It is fully extensible and will no doubt incorporate further learning concepts as the E-Learning experiment continues. It works in the following way: as students assign values to array elements the array can be seen visually on the screen. Only an array up to 8 wide and 8 high can be displayed onscreen although larger arrays can be declared and populated. The prototype is standalone, and does not record or monitor a student's performance. Instead it focuses on the repetition of entering code, correction of errors and visualization of results of the program entered. The prototype GUI consists of a single screen that does not change and is sub-divided into 3 areas:

Area 1. Displays an 8 * 8 grid of the created array with the values stored in each element.

Area 2. As each line of code is executed a "feedback" message is displayed indicating what action is being performed.

Area 3. Area where user enters their Java code to create and populate the array. Basic controls to Step through a users code, execute a users code or create a new program are also provided.

5. Survey results and Analysis

This section provides an analysis of the survey results to attempt to draw conclusions as to the effectiveness of the E-Learning prototype and any contributions it may have made to enrich the learning experience of the students. The questionnaires were designed to ascertain the general group level of knowledge and understanding within each distinct Array concept. The results of these groups can then be compared against each other to determine what effect if any the E-learning tool had on the student group who used the prototype. The tool may have had a positive, negative or null effect on the students learning experience.

Before the study began, the previous performance of the two classes was noted by analysing the results of the practical in lab Continuous Assessments that they had recently completed and being graded on. The calculations used the average results that were gained and average performance of each group is summarised below.

	Group1 (No Prototype)	**Group2 (With prototype)**
Assessment # 1	74.47%	61.85%
Assessment # 2	62.78%	58.38%
Overall Average %	68.63%	60.12%

This average percentage was used to determine a relative performance difference between the two groups initially, so that this difference could be taken into account after the study had been completed.

The first questionnaire was given to the students after the concept of array declaration and initialisation had been covered. The questionnaire contained a set of multiple choice (closed questions) and open-ended questions such as

> *"Write code to declare and initialise a 5 element integer array called myArray"*
> *"Describe what an 'Array' is in your own words?"*

The second and third questionnaires were distributed during and after the concept of array manipulation. These questionnaires also contained sets of multiple choice and open-ended questions such as

"What is the output to the screen if the following code is executed"

```
int [] MyArray = {1,2,3,4,5};

for (int i=0;i<MyArray.length;i++)
{
        System.out.print(MyArray[i] + " ");
}
```

The fourth questionnaire also followed the same style design as the first, second and third questionnaires, but focused on Multi-Dimensional Arrays.

Each set of questionnaires was assessed and graded for both groups and an average percentage based on knowledge and performance was calculated for each group. The average percentage based on each questionnaire for each group is summarised below.

	Group 1 (No Prototype)	**Group 2 (With Prototype)**
Array Initialisation	96.87%	85.73%
Array Manipulation #1	75.45%	80.68%
Array Manipulation #2	60.76%	43.85%
Multi-Dimensional Arrays	73.33%	65.71%
Average Group performance	76.61%	69.01%

The fifth and final questionnaire focused entirely on the E-Learning prototype itself and was only distributed to the group that had been using the prototype during this time. This questionnaire consisted primarily of closed questions, which attempted to ascertain any positive or negative qualities the students liked or disliked about using the prototype. These questions asked the students for their own opinions as to how the prototype affected their learning experience.

"Did the tool help you 'create' your own arrays?"
"Did you 'like' using the E-learning tool"?

A substantial amount of student feedback was gathered, based on this final questionnaire. All students who returned a completed questionnaire felt that the prototype had helped them to

increase their basic knowledge of arrays. When asked to explain how the prototype had increased their knowledge and understanding, students replied with comments such as

> *"It makes it easy to see where things are coming from"*
>
> *"It actually shows you what happens and where it happens. It makes it easier when you physically see what happens"*
>
> *"The ability to troubleshoot makes it easier to identify mistakes & learn from them."*

Students also felt that the prototype helped *explain* the concept of arrays by allowing the students *visualise* concepts graphically

> *"…makes the purpose of arrays more obvious"*
> *"…it's easier to explain as I have used it"*
> *"…much better understanding of how elements are stored in an array"*

Students also felt that the prototype provided feedback *"clearly and accurately"* through an interface that was *"well designed and easy to use"*. They also felt that without the prototype they would have been *"trying to figure it out on paper"* and that *"…if we had just used the Java editor (Forte), I wouldn't have understood what was going on, this tool made it much clearer"*.

Some of the feedback suggested that the prototype could have been improved to provide better feedback or to include a demonstration by example. One student stated, *"…it only tells you there is an error but didn't tell you how to fix it or the solution to the problem"*. Students felt that the prototype was *"motivating"* and if it was made available on the Departments Intranet facility (Compweb) that they would use the prototype again, *"...because of its capability to make learning arrays easier"*

6. Conclusions

The aim of the study was to determine if an E-Learning prototype would make any significant contributions to novice students learning arrays in IT Tallaght. The results although tentative are as follows.

Prior to the survey, the group that did not have the prototype scored in performance an average of 8.5% more than the students who were given the prototype. After the controlled study, this group still scored more, this time an average of 7.6% greater. This implies that the group with the prototype very marginally closed the performance gap, but the percentages calculated are too marginal to attribute any "real" positive contribution to learning to the E-learning prototype. Therefore, the prototype in this study did not make any measurable quantitative impact in this particular study. However, where it did make a sizeable impact was in the qualitative study, which focused on the class who used the prototype and attempted to ascertain whether they saw any value in using the prototype as a learning tool. Based on feedback from these students, they felt more "engaged", "motivated", found it "easier to visualise arrays" and they "enjoyed" using the prototype as it provided an alternative approach to applying these concepts through repetition and gave instant visual feedback in response to Java code statements.

The primary but tentative conclusion that has been reached as a result of this study is that the prototype did not demonstrate any real measurable academic improvement in the student's knowledge or application of "Arrays in Java" but it did positively enrich the students learning experience. This warrants further investigation with a rationale to further develop a more advanced E-Learning prototype in this "Java Arrays" area, which will incorporate a more comprehensive instructional design methodology in its design and development.

Hardware/Software Codesign

Richard Gallery & Deepesh M. Shakya
School of Informatics and Engineering, ITB
{Richard.Gallery,Deepesh.Shakya@itb.ie}

Introduction

The current state of the art technology in integrated circuits allows the incorporation of multiple processor cores and memory arrays, in addition to application specific hardware, on a single substrate. As silicon technology has become more advanced, allowing the implementation of more complex designs, systems have begun to incorporate considerable amounts of embedded software [3]. Thus it becomes increasingly necessary for the system designers to have knowledge on both hardware and software to make efficient design trade-offs. This is where hardware/software codesign comes into existence.

Hardware/software codesign is the concurrent design of both hardware and software of the system by taking into consideration the cost, energy, performance, speed and other parameters of the system. During the design, trade-offs are made between the implementation of functionality in hardware and/or software depending upon both cost considerations and technical feasibility.

Since the concept of hardware/software codesign surfaced in 1990s [1], different methodologies have been proposed for hardware/software codesign. This article gives an overview of hardware/software codesign. In section 2, a generic hardware/software codesign methodology is described, section 3 describes the taxonomy of hardware/software codesign where different aspects of hardware/software codesign is discussed along with some works performed in these arena to date, section 4 gives an introduction of different codesign methodologies widely accepted in the literature.

Generic Hardware/Software Codesign Methodology

In this section a generic methodology for hardware/software codesign (
Figure ***1***) is discussed. The initial step in hardware/software codesign is the high level specification of the system behaviour to include the functionality, performance, cost, power and other constraints of the expected design. The specification step includes modelling of the system in order to capture the entire characteristics of the system.

After the system specification is ready, it is divided into a number of blocks to allow a costing, through the application of cost metrics[2] for each of these blocks. This is performed in the Cost Estimation step where the estimation is done for both hardware and software implementation. This is actually the step for analysis and estimation. The system is analysed from different aspects of its cost metrics. This step provides valuable information for the hardware/software partitioning.

The next stage is to partition the system functionality between hardware and software. The partitioning phase takes information collected from the Cost Estimation phase to allow decisions to be taken as to which block is to be mapped on hardware and which block to be mapped on software. The quality of such mapping depends on how much the design constraints have been

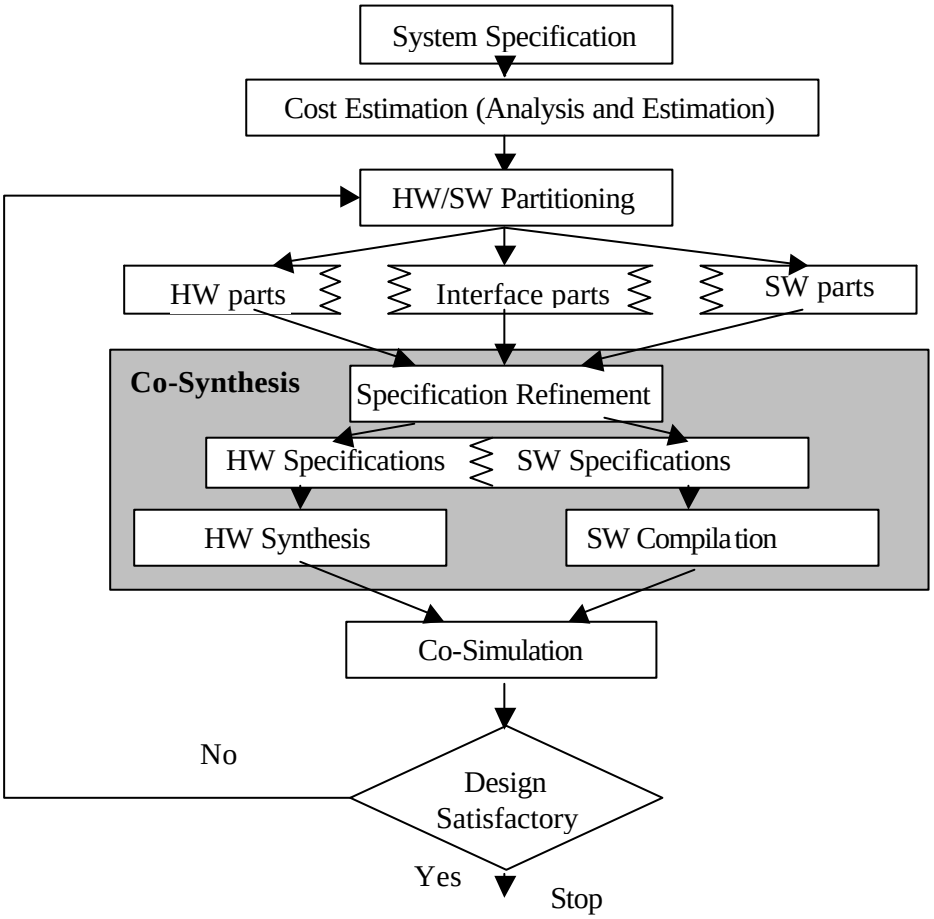


Figure 1 Generic Hardware/Software Codesign Methodology [2]

[2] Cost metrics can be calculated for both hardware and software. Hardware cost metrics can be, for e.g., execution time, chip area, power consumption or testability. Software cost metrics can be, for e.g., execution time, program and data memory.

achieved and how much the design cost is minimized [3]. If the design constraints[3] are not met then the expected performance of the system cannot be met and if the design cost is not minimized, the design cannot compete in the market.

The system is then designed within the context of the heterogeneous target architecture[4]. This requires the specification of the interfaces (communication and synchronization) between hardware represented by ASICs and software represented by the processors.

Once hardware/software blocks and the interfaces between them have been decided, the next step is the co-synthesis. In co-synthesis, specification refinement is done, where the implementation independent system specification is transferred into hardware and software specifications (and the specifications for the interfaces).

Once the separate specification and the necessary refinement[5] in the hardware design are carried out, hardware is synthesised that gives a set of ASICs and software is compiled for the target processor.

Once the synthesis step is over, the next step in the design flow validates the system design simulating both the ASIC and the processor together. This is called the co-simulation. The co-simulation checks whether the design goal has been achieved or not. If the design is acceptable, the codesign flow stops. If the design is not acceptable the design is traced back to the hardware/software partitioning step and the design cycle is repeated until the satisfactory design output is obtained [2].

Taxonomy of Hardware/Software Codesign

Hardware/Software has the following important aspects [3] which must be considered for an effective system design.

- Modelling
- Analysis and Estimation
- System Level Partitioning, Synthesis and Interfacing
- Implementation Generation
- Co-Simulation and Emulation

[3] There can be different design constraints for e.g. time, area, power, memory etc. Timing constraints specifies timeline for the execution of the system task.
[4] System architecture consisting of both hardware (ASICs) and software (general processor).

Modelling

Modelling can be regarded as the science of capturing the system characteristics [3]. Models should capture all the information which is necessary for the designers.

Edwards et al.[4] explores the various computational models in the embedded system design, in which they stress that the formal model should include:

- Formal specification (relation between input and output and internal states),
- Set of properties[6] (a set of relation between input and output and internal states. This is explicitly mentioned to verify it against the functional specification. The properties are for assertion of the behavior rather than description of the behavior.)
- Performance indices (e.g. cost, reliability, speed, size etc.)
- Design constraints (on performance indices).

The functional specification fully characterizes the system while satisfying the set of properties.

The design process starts by modelling the system at a high level of abstraction. The designer checks whether a set of properties have been verified, performance indices are satisfactory and the design constraints are met.

Edwards et al.[4] goes on to recognize four different types of Models of Computation[7]:

- Discrete Event,
- Communication Finite State Machines (FSM)
- Synchronous/Reactive
- Dataflow Process Networks Models.

A discrete event (DE) model is characterized by events which are time-stamped (i.e. the time at which event occurs is timestamped). A DE simulator requires a global event queue which keeps track of the time-stamped events and orders them to be executed according to their time-stamps. The DE approach is used to simulate the digital hardware [5].

---

[5] An act of adding design details or converting from abstract representation to Register Transfer Level (RTL) ready to be fed into the hardware synthesis tool.

[6] It can be property of determinate behavior i.e. the output of the system depends entirely on its input and not on some internal hidden factors.

[7] A system can be thought of as composing of simpler subsystems, or pieces. The method or the rules for composing and capturing these pieces to create a system functionality is called models of computation.

FSMs are good for modelling sequential behaviour but are not suitable for modelling concurrent behaviour of a group of machines, as it may reach the point state explosion[8]. This is because the number of states will be the product of the number of states in each machine.

The synchronous/reactive model consists of events which are synchronous i.e. all signal have events at clock tick. The simulators that use the synchronous models are called cycle-based or cycle-driver simulators.

In the dataflow model, there is a directed graph where the nodes represent computations and ordered sequences of events which is represented by arcs [6].

The above discussed modelling techniques may be deployed through the use of appropriate modelling or specification languages. For example designers using SDL[9] (a state oriented language) [7] can describe multiple concurrent processes[10] which communicate with signals. StateCharts [8] has the ability to decompose a sequential model into hierarchical structure thus facilitating the representation of FSMs in Statecharts. This hierarchical decomposition can solve the problem of state explosion [11][3]. The Esterel [9] language is a synchronous language that supports concurrent behaviour and thus makes it suitable for modelling in FSMs [9]. SpecCharts [73] exploits the advantage of hierarchical and concurrent state diagrams and the hardware description language VHDL [61].

Specification languages

Specification language describes the overall goal of the desired functionality of a system. A good specification language is able to address different aspects of a system which includes following [2] [52] [54]:

- Concurrency
- State-transitions
- Hierarchy

[8] The number of state grows exponential that makes the design complex enough to be handled.
[9] Specification and Description Language
[10] Process is a program codes consisting of sequence of statements.
[11] States contained within a state are called sub-states of this surrounding state. The surrounding state is higher in the hierarchy. If there are two machines with 4 states each then if these machines are combined to form a single machine then the total number of states of the combined machine will be the permutation of the number of states of each machine. If in any case, all the states (both machines combined) can be arranged in an hierarchical manner for e.g. all four states of one machine can be considered as substates of one of the state of another machine then these substates will have nothing to do with other states of the machine hence the total number of possible number of states is reduced.

- Programming constructs
- Behavioral completion
- Communication
- Synchronization
- Exception Handling
- Non-determinism
- Timing

Concurrency: Parts of an embedded system work in parallel. Parts may be a process and threads of the process. A specification language should be able to capture the concurrency[12] behavior of the system.

State-transitions: Systems are often conceptualized as having various modes or states, of behavior. In a system with multiple states, the transition between states occurs in undefined or unstructured manner. The specification language must be able to model such arbitrary transitions.

Hierarchy: A system can be conceptualized as a set of smaller subsystems if modelled as hierarchical models. Such conceptualization helps system designer to simplify the development of a conceptual view of a system, since parts of the system can be treated as a separate unit paving way for scoping objects, such as declaration types, variables and subprogram names. Lack of hierarchy will make all such objects global and it becomes increasingly difficult for a designer, as the system will become more complex. There are two types of hierarchy: structural hierarchy and behavioral hierarchy. *Structural hierarchy* enables designer to design a system with interconnected components. Each component is themselves a composition of sub-components. *Behavioral hierarchy* decomposes a system behavior into distinct sub behaviors for e.g. procedures or functions.

Programming Constructs: Specification language should have programming constructs, for e.g. constructs like functions, procedures, loops, branches (if, case) and assignments simplifies the sequential description of the system behavior.

Behavioral Completion: A specification language should be able to model the behavioral completion to indicate that the behavior has completed the execution of all computations. An advantage of behavioral completion is that it allows designer to conceptualize the behavior as an independent module. The designer may start next behavior in sequence once the preceding behavior has finished without worrying if any unfinished work remained in that behavior.

Communication/Synchronization: A system has several processes working concurrently. These processes need to communicate with each other. A specification language should have an ability to model the communication between concurrent behaviors or processes and at the same time should ensure the synchronization of two behaviors or processes. Such communication can be conceptualized as a shared memory[13] or message passing[14] paradigms. In shared memory, the sending process writes to a medium which is also accessible to the receiving process. Such a medium could be global variables or ports. In message-passing communication model, the communication is accomplished with an abstract medium called channels with send/receive primitives.

Exception handling: A specification language should be able to model the exception handling mechanism, for e.g. when an exception occurs in the form of interrupts or resets, the current state of the system should be terminated and the transition to the new state is required. Such reactive behavior is quite common in the embedded system.

Non-determinism: Non-determinism is helpful when the designer doesn't want to take decision during the specification, for e.g. if two events occurs simultaneously then the designer can leave the decision of which event to be executed first at the time of implementation. This is only possible if the specification language has an ability to model the non-determinism.

Timing: Specification language should have an ability to model the timing aspects of the embedded system which are the functional timing and the timing constraints. *Functional timing* represents a time required to execute a behavior. *Timing constraints* indicate a range of time within which a behavior has to be executed.

The specification languages have been categorized into the following categories as presented in [2].

1. Formal Description Technique (for e.g. LOTOS [56], SDL [55], Estelle [57])
2. Real Time System Languages (for e.g. Esterel [58], Statecharts [59], High-Level Time Petri Nets [60])

---

[12] The act of two processes running concurrently.

[13] Memory in a parallel computer, usually RAM, which can be accessed by more than one processor, usually via a shared bus or network. It usually takes longer for a processor to access shared memory than to access its own private memory because of contention for the processor-to-memory connections and because of other overheads associated with ensuring synchronised access.

[14] A message passing system has primitives (for e.g. send ( ) and receive ( )) for sending and receiving messages. These primitives can be either synchronous or asynchronous. In synchronous message passing, sending and receiving of message is not complete unless receiving end acknowledges the receipt of the message. In asynchronous message passing, the message sending process in complete once the message is sent irrespective of whether the message has been received by the receiving end or not.

3. Hardware Description Languages (for e.g. SpecCharts [73] [54], VHDL [61][61], Verilog [62], HardwareC [44], Handel-C[70])
4. Programming Languages (for e.g. SpecC [63] [64] [65] ,C^x [19])
5. Parallel Programming Languages (for e.g. CSP [66], Occam [67])
6. Data Flow Languages (for e.g. Silage [68] [2])
7. System Design Language (for e.g. SystemC [69])

Analysis and Estimation

It becomes necessary for designers to take crucial decisions during the codesign process and in order to take these decisions a designer requires:

- Application domain knowledge (ideally the designer understands the application domain in which the technology will be deployed)
- Knowledge of the technology options that are available
- The ability to analyse proposed design solutions (and as a result access to, training and knowledge of the capabilities and limitations of design tools)

The analysis and estimation of the design become more crucial when the design constraints require fast timing and high power consumption [3]. Correct procedures in the design process can avoid non-competitive and costly designs.

There are different analysis types that need to be made in the design [3], including, amongst others:

- Process path analysis
- Architecture modelling and analysis
- Power analysis
- Multiprocessor analysis rate analysis
- memory system analysis

A process path is a sequence of process statements that is executed for given input data [3]. *Process-path analysis* corresponds to determining a set of possible process paths. By examining the possible process paths, it is possible to find out the worst case execution time (WCET) by first tracing the process paths during worse case and then calculating the WCET. Li et al. extensively discusses the process path analysis in [10].

Power analysis consists in determining the power cost of the system. Tiwari et al. in [11] describes a power analysis technique to generate the power cost model for the embedded software. Fornaciari et al. in [12] introduces power metrics included in a hardware/software codesign environment to guide the system level partitioning. Yanbing Li et al. in [13] explores a framework for optimizing the system parameters to minimize energy dissipation of the overall system i.e. both hardware and software. The paper also explores the trade-off between system performance and the energy dissipation.

Rate analysis includes the analysis of execution rate of the processes. The rate constraints, imposed by the designer in order to assure proper working of the system to its environment, is one form of the timing constraints [14]. Mathur et al. in [14] proposes an interactive rate analysis framework to make sure all the rate constraints are satisfied in the design of the system.

Memory system analysis is also an important factor in the embedded system design. Specially, in the areas of image and video processing systems, 50-80% of area cost of the ASICs for real-time multidimensional signal processing is due to the data storage and transfer of array signals [15]. So it becomes increasingly necessary to estimate the memory usage and optimize them well before any decision on hardware software partitioning is made.

Multiprocessor analysis deals with the estimation and analysis of parallel process execution. Process scheduling decides the order of process execution.

System-Level Partitioning, Synthesis, and Interfacing

Partitioning

Hardware/software partitioning takes place after the necessary information on cost metrics is generated from the analysis and estimation of the design. Based upon this information, the system is divided into hardware and software according to whichever gives the best overall performance result.

Various algorithms have been developed for the hardware/software partitioning. Gupta and DeMicheli [16] [17] created an algorithm to automate a search of the design space for the hardware/software partitioning. The algorithm starts by implementing all functionalities in hardware and the operations to be moved into software are selected based on the cost criterion of communication overheads. Movement into software is only done if there is any

improvement in the cost of the current system partition. The algorithm iterates the process of movement until no cost-improving move could be found. The main defect in this algorithm is that the algorithm frequently created very costly hardware that consumes many resources, since the initial partition starts with the hardware solution [18]. Authors in [17] depict the use of their algorithm for describing the implementation of a network coprocessor communication via an ethernet[15] link. This co-processor is used to take load off the CPU to handle the communication activities.

Ernst and Henkel [19] start with the initial partition in software and gradually transfer the software part into hardware. Ernst and Henkel used a hill-climbing[16] partitioning heuristic, an example of which is the simulated annealing [53]. This algorithm uses a cost function to minimize the amount of hardware used with the performance constraints remaining intact. Simulated annealing in [19] starts with an infeasible solution with a high cost penalty for run time exceeding timing constraints. Then the algorithm searches for an improved timing and a steep decrease in the cost. Ernst and Henkel in [19] uses their algorithm for the hardware/software partitioning of the digital control of a turbocharged diesel engine and a filter algorithm for a digital image in which they got a speed up of *1.4* and *1.3* respectively in reference to the implementation in software alone.

Synthesis and Interface

Once the partitioning specification is ready, the next step is the synthesis of hardware, software and their respective interfaces. In other words, the co-synthesis step follows next after the hardware software partitioning. Co-synthesis is defined as the synthesis of hardware, software and the interface between hardware and software. Once the synthesis is complete then the design is subjected to co-simulation.

The final synthesized system architecture generally comprises of: a programmable processor, one or more hardware modules all of which are connected through a system bus, and the appropriate software modules and interfaces. Hardware modules consist of a datapath, a controller and I/O interface between hardware and the processor. The processor runs the

[15] Ethernet is a physical and data link layer technology for LAN networking.

[16] Hill-climbing algorithms are neighborhood search algorithms that subsequently select the neighbor with the highest quality and continue from there. The search terminates when no neighbor exists that represents an improvement over the current solution. Hill-climbing algorithms belong to the class of greedy algorithms i.e. the algorithm never goes back to a solution with lower quality. In other words, the climber never goes downhill to finally reach a higher peak [55].

software component of the architecture and also includes the device drivers to establish communication between software and hardware (the hardware/software interfaces).

In [20], an environment is described for the specification and the synthesis of a heterogeneous system using Cosmos[17]. Design starts with an SDL[18] [47] specification and produces a heterogeneous architecture comprising hardware in VHDL and software in C. Codesign steps in Cosmos includes: partitioning, communication synthesis and architecture generation. Communication synthesis consists in transferring the process that communicates with high-level primitives[19] through channels into signals. Architecture generation here is actually an implementation generation step discussed in the next section. Architecture generation includes two major tasks i.e. virtual prototyping and architecture mapping. Virtual prototyping consists of hardware (in VHDL), software (in C) and communication (in VHDL or C) which can be simulated. Architecture mapping consists of synthesizing VHDL descriptions into the ASICs, conversion of software parts into assembly code resulting in the final architecture that consists of software, hardware and the communication components.

The software design and the software synthesis are also an important aspect of the hardware/software codesign since a significant part of the system (i.e. the system that consists of both hardware and software) are implemented in software. Software synthesis focuses on the support of embedded systems without the use of operating systems[20] [21].

Implementation Generation

Implementation generation for hardware refers to generating hardware for a set of functions. Hardware typically consists of [24]:

- Control-unit/datapath
- Storage unit (for e.g. registers, register files and memories)
- Multiplexer

[17] Cosmos is a co-design methodology and tools aimed at the design and synthesis of complex mixed hardware-software systems.
[18] Specification Description Language.
[19] Primitives are the basic operations. High level primitives for the communication between two processes can be taken as the communication that occurs by calling functions.
[20] The main drawback of using the support of operating system is that most kernels tend to use a fixed priority preemptive scheduling mechanism, where the timing constraint is realized from the process priorities. In some cases the timing constraint is realized by scheduling the process with information of process period, release time and deadline. But in embedded system, the timing constraints should be realized more on the occurrence of the events. Since, the operating system scheduling mechanism doesn't have idea on the time stamp; it doesn't know when the events are generated. [21]

- State-register
- Control-logic

While generating hardware, the size of hardware should be as small as possible while maintaining the system constraints intact. An implementation, which is silicon area efficient, is thus a sign of quality design. Implementation generation for software refers to producing an assembly code for software. An efficient software implementation can only be realized if the compilers can take full advantage of the architectural features of the processor. Some approaches for exploiting architectural features are described below.

Sudarsanam et al. in [23] presents a retargetable[21] methodology in an effort to generate high quality code for a wide range of DSPs[22]. The paper describes a solution for the problems arising in those compiler technologies which are unable to generate dense, high-performance code for DSPs as they do not provide adequate support for the specialized features of DSPs. Also, the paper describes the solution for the problem where it is necessary to build a compiler from scratch, due to the unavailability of a suitable compiler (a time consuming process). The solution presented is a methodology for developing retargetable DSP compilation.

Vahid et al. in [24] describes an algorithm for estimating the hardware size. The paper describes an algorithm for the hardware estimator, which is based on incrementally updating the design model to acquire accuracy and iterative improvement algorithms to explore the different design possibilities. Hence, the algorithm maintains both speed and accuracy in estimating hardware size. The algorithm takes advantage of the fact that between two iterations of partitioning design there is only an incremental change. For this incremental change, a data structure (representing an incrementally modifiable design model) and an algorithm that can quickly provide the basic design parameters needed by the hardware-size estimator are developed. Therefore, whenever there is any incremental change in the design model, the corresponding hardware size is estimated.

[21] Retargetable means the reuse without little or no modification for e.g. retargetable compiler is able to generate code (maintaining the same quality) for the new processor after minor modifications without need of creating entirely new compiler from the scratch.
[22] Digital Signal Processor

Co-Simulation and Emulation

Co-simulation

Co-simulation of hardware and software refers to the simultaneous verification of hardware and software functions correctly [25]. The conventional co-simulation approach waits until the real hardware has been delivered and then performs verification by using in-circuit emulators[23]. Due to the increased complexity of the designs and the importance of verifying the system design as much as possible before committing to the (expensive) transfer of the hardware aspects of the system to silicon, it has become necessary to perform co-simulation before the real hardware is produced. This saves time-to-market as well as the cost required in debugging and re-building the hardware. Rowson in [25] gives an overview of the techniques available for hardware/software co-simulation.

Ghosh et al. in [32] describes a hardware-software co-simulator that can be used in the design, debugging and verification of embedded systems. This tool consists of simulators for different parts of the system (for e.g. Clock Simulator, Parallel Port Simulator, UART simulator, CPU Simulator, Timer Simulator, Memory Simulator etc.) and a backplane[24] which is responsible for integrating all the simulators. The back plane is represented by Simulation Manager which manages communication between the co-simulators (e.g. CPU simulator, Memory Simulator etc.) and the virtual instruments. Virtual instruments are used to provide stimulus and to observe response. The paper describes a tool that provides an environment for joint debugging of software and hardware and is also capable of evaluating system performance, selection of algorithms and implementations. The tool also addresses the possibility of exploring hardware-software tradeoffs.

In [32], the performance of the tool for the applications (which was taken as an example) like engine control unit has been evaluated. The co-simulation of the engine control unit showed a slowdown by a factor of 400 which is quite suitable for debugging.

Valderrama et al. in [33] describes a unified co-synthesis and co-simulation methodology i.e. both the steps are performed using the same descriptions (in C and VHDL). The

[23] In-circuit emulators are used to replace the processor or microcontroller of the target hardware. It is a valuable software developers tool in embedded design. The developer loads the program into the emulator and can then run, step and trace into it.

[24] Simulation backplane controls all simulators coupled to it. If a simulator needs to communicate with partner-simulators, it does this through the simulation back plane.

communication between hardware and software is through a communication unit, which is an entity able to execute a communication scheme invoked through a procedure call mechanism [74]. The VHDL entity[25] is used to connect a hardware module with that of software. The use of procedure call mechanism hides the implementation details related to the communication unit. The access to the interface of the communication is done through the procedures. By employing this method, the two communicating modules become quite independent of each other and changes in one module need not change in other module unless the communication unit interface is being accessed using the same procedure before and after the change. The level of abstraction obtained by using procedures help in using the same module descriptions with different architectures (i.e. the architectures which varies depending upon the communication protocols used).

Emulation

Co-simulation uses an abstract model to form a virtual prototype (of hardware) while co-emulation provides a real prototype by implementing functions in hardware (for e.g. FPGA[26]).

Luis et al. in [34] describes the co-emulation process observed in the co-design methodology-LOTOS [56]. Once all the construction of the interface between hardware and software is completed, the execution of software (in C) and the simulation of hardware (in VHDL simulator) is performed on SUN workstation. The things that software requires to write or read into or from the FPGA (i.e. the hardware) is written into the files (through C functions) and the hardware simulator reads the files via an extra VHDL component in the interface, which is a wrapper[27] for a set of C functions that perform reading and writing operation on the files. This is to perform a test for errors before emulating hardware with the FPGA. The next step performed is the co-emulation where the hardware part is replaced by the FPGA.

Cedric et al. in [71] describes the co-simulation between SystemC and an emulator called ZeBu. The paper depicts how SystemC can be co-simulated with ZeBu at different level of abstraction i.e. at signal-level and at transaction level[28]. ZeBu is a hardware verification product built on a PCI card with Xilinx Virtex-II FPGA [72] devices. ZeBu consists of a technology called Reconfigurable Test Bench (RTB) that interfaces a design under test

[25] A modular representation of deign in VHDL is called an entity.
[26] Field Programmable Gate Array
[27] Wrapper is a piece of code which is combined with another code to determine how the latter code is executed. Wrapper actually acts as an interface between its caller and the wrapped code.
[28] The communication that takes place with function call.

(DUT). DUT is emulated by one or more Virtex-II FPGA devices. The main function of the RTB is to stimulate and monitor each individual I/O data pin of the DUT emulated by the FPGA. ZeBu also consists of C/C++ API which works in concert with the RTB providing direct interaction with the test benches modelled at higher level of abstraction via SystemC. In the paper, a test case is presented in which the co-simulation for a graphics design is conducted for three different cases: SystemC model and HDL[29] (Verilog), SystemC model and ZeBu at the signal level and SystemC model and ZeBu at the transaction level. SystemC models consist of test bench that interacts with the emulated hardware. The result shows that the co-simulation execution time for SystemC/HDL is *3* days (for that particular test case considered), SystemC/ZeBu at signal level is *330* seconds and SystemC/Zebu at the transaction level is *5* seconds. This co-simulation process with emulated hardware is one of the latest technologies in the literature. The main benefit of this technique is its ability to co-simulate at transaction level that gives significant speed-ups.

Co-design Systems

In section 0, a generic hardware/software codesign methodology was presented. In this section different codesign approaches will be introduced. An interested reader on particular codesign system may refer to the references given against a methodology name introduced here.

Ptolemy [35] is codesign methodology that allows heterogeneous specification[30] to develop a unified environment for creating heterogeneous systems. Castle [36] [37] is a codesign platform which puts more emphasis on processor synthesis i.e. starting from an application it ends up with synthesis of suitable processor design on which the application considered can be executed efficiently. Cosyma[31] [38] is a codesign methodology which starts the system solution all in software and during the partitioning step gradually ports software portion into hardware to achieve practically feasible design. Lycos[32] [39][40] is a codesign tool based on a target architecture with the single processor and a single ASIC. Lycos stresses design space exploration[33] with automatic hardware/software partitioning. Tosca[34] [22][41] is a codesign

[29] Hardware Description Language to simulate the hardware.
[30] System specification with more than one specification language.
[31] Co-synthesis for Embedded Architectures
[32] Lyngby Cosynthesis
[33] Choosing one suitable design out of many.
[34] Tools for System Codesign Automation

methodology that is mainly targeted for control flow dominated[35] reactive real-time systems [2]. The target architecture in Tosca consists of off-the-shelf processors and a set of co-processors on a single chip. The design description is specified using C, VHDL or Occam [67]. Vulcan [42][43] is a hardware/software codesign tool focusing on the co-synthesis. The input specification to this codesign tool is the hardware description language, HardwareC [44]. The partitioning in Vulcan starts with a complete solution in hardware i.e. describing the entire solution in HardwareC [44]. Chinook [45] is a co-synthesis tool for embedded real time systems. Chinook focuses on the synthesis of hardware and software interface and communication. Cosmos [46] is a codesign environment in which the system description is specified in SDL[36] [47] and ends up by producing a heterogeneous architecture with the hardware descriptions in VHDL and the software descriptions in C. CoWare [48][2] is a codesign environment of a system supporting co-specification (heterogeneous specification), co-simulation and co-synthesis (heterogeneous implementation). Polis [49] is a framework for hardware/software codesign targeted for the reactive embedded systems[37]. The system is specified in the specification language called Esterel [50]. SpecSyn [51] is a codesign environment which supports a specify-explore-refine design paradigm i.e. the design starts with the specification of the system functionality and then the rapid exploration of numerous system level design options is performed. Once the feasible most option is selected then refinement is carried out for the chosen option

Conclusion

Hardware/software codesign is relatively a new topic but since its inception, its literature has grown to a wide range of arena and many researches have been conducted in this field. There is no standard co-design methodology which can be regarded as the most useful for every system design. All the methodologies that are available in the literature till date has its own advantages and disadvantages. In some cases, it only suits specific applications. Competitive product with low cost and less time to market is the manifestation of an efficient design methodology. However, methodology alone is not sufficient; it needs equally strong specification language, suitable model of computation, efficient compiler, efficient synthesis tool and the efficient co-simulation environment.

[35] System which is determined at run time by the input data and by the control structured (e.g. "if" statements) used in the program.
[36] Specification and Description Language
[37] Reactive systems typically respond to incoming stimuli from the environment by changing its internal state and producing output results [2].

References

[1]. Wayne Wolf, *A Decade of Hardware/Software Codesign*, Article from Computer, pp. 38-43, April 2003.

[2]. Ralf Niemann, *Hardware/Software Co-design for Data Flow Dominated Embedded Systems*, Kluwer Academic Publishers, 1998.

[3]. Jorgen Staunstrup and Wayne Wolf, *Hardware/Software Co-Design: Principles and Practice*, Kluwer Academic Publishers, 1997.

[4]. S. Edwards, L. Lavagno, E.A. Lee, and A. Sangiovanni-Vincentelli, *Design of Embedded Systems: Formal Models, Validation, and Synthesis*, Proc. IEEE, vol. 85, pp. 366-390, Mar. 1997.

[5]. V. Lazarov and R. Iliev, *Discrete Event Simulation of Parallel Machines*, 2nd AIZU International Symposium on Parallel Algorithms / Architecture Synthesis, pp. 300, March 1997.

[6]. Edward A. Lee and Thomas M. Parks, *Dataflow Process Networks*, Proceedings of the IEEE, vol. 83, no. 5, pp. 773-801, May, 1995.

[7]. M. Daveau, G. Marchioro and A. Jerraya, *VHDL generationfrom SDL specification*, In: CHDL. pp. 182-201, 1997.

[8]. Harel, D., *Statecharts: A Visual Formalisms for Complex Systems*, Communications of the ACM Vol.31 No.5, 1988.

[9]. G. Berry and G. Gonthier, *The Esterel synchronous programming language: Design,semantics, implementation*, Science Of Computer Programming, 19(2):87–152, 1992.

[10]. Yau-Tsun Steven Li and Sharad Malik, *Performance analysis of embedded software using implicit path enumeration*, Proceedings of the 32nd ACM/IEEE conference on Design automation conference, USA, 1995.

[11]. Vivek Tiwari, Sharad Malik and Andrew Wolfe, *Power analysis of embedded software: a first step towards software power minimization*, IEEE Transactions on VLSI Systems, December 1994.

[12]. William Fornaciari , Paolo Gubian , Donatella Sciuto and Cristina Silvano, *Power estimation of embedded systems: a hardware/software codesign approach*, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, v.6 n.2, p.266-275, June 1998.

[13]. Yanbing Li , Jörg Henkel, *A framework for estimation and minimizing energy dissipation of embedded HW/SW systems*, Proceedings of the 35th annual conference on Design automation conference, p.188-193, California, United States, June 15-19, 1998.

[14]. A. Dasdan, A. Mathur, and R. K. Gupta. *RATAN: A tool for rate analysis and rate constraint debugging for embedded systems*. In Proceedings ED&TC '97, 1997.

[15]. Koen Danckaert, Francky Catthoor and Hugo de Man, *System level memory optimization for hardware-software co-design*, 5th International Workshop on Hardware/Software Co-Design (Codes/CASHE '97) Braunschweig, GERMANY, March 24 - 26, 1997.

[16]. R. Gupta and G. De Micheli, *Hardware-software cosynthesis for digital systems*, IEEE Design and Test of Computers, vol. 10, no.3, pp.29-41, Sept. 1993.

[17]. R.K. Gupta and G.D. Micheli, *System-level Synthesis using Re-programmable Components*, 1EEE/ACM Proc. of EDAC'92, 1EEE Comp. Soc. Press, pp. 2 -7, 1992.

[18]. Adam Kaplan, Majid Sarrafzadeh and Ryan Kastne, *A Survey of Hardware/Software System Partitioning*, (Details not available)

[19]. Rolf Ernst, Jorg Henkel and Thomas Benner, *Hardware-Software Cosynthesis for Microcontrollers*, Design and Test of Computers, pp. 64-75 ,Vol. 10, No. 4, October/December 1993.

[20]. Tarek Ben Ismail, Ahmed Amine Jerraya, *Synthesis Steps and Design Models for Codesign* ,Computer, Vol. 28, No. 2, pp44-52, February 1995.

[21]. Filip Thoen, Marco Cornero, Gert Goossens and Hugo De Man, *Real Time Multi-Tasking in Software Synthesis for Information Processing Systems*, Eighth International Symposium on System-Level Synthesis, Los Alamitos, 1995.

[22]. A. Balboni, W. Fornaciari, and D. Sciuto, *Co-synthesis and cosimulation of control dominated embedded systems*, in International Journal Design Automation for Embedded Systems, vol. 1, no. 3, July 1996.

[23]. Ashok Sudarsanam, Sharad Malik and Masahiro Fujita, *A retargetable Compilation Methodology for embedded Digital Signal Processors using a Machine-Dependent Code Optimization Library*, Design Automation for Embedded Systems, Kluwer Academic Publishers, pp. 187-206, 1999.

[24]. Frank Vahid and Daniel D. Gajski, *Incremental hardware estimation during hardware/software functional partitioning*, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol.3 No.3, pp.459-464, Sept. 1995.
[25]. J. Rowson, *Hardware/software co-simulation*, In Proc. of the Design Automation Conference, pp. 439--440, 1994.
[26]. Heiko Hubert, *A Survey of HW/SW Cosimulation Techniques and Tools*, Thesis work, Royal Institute of Technology, Sweden, 1998.
[27]. Triscend, *Bus Functional Model*, Application Note,(AN_32),v1.4, July 2001.
[28]. Virtutech White Papers, *Introduction to Simics: Full System Simulator Without Equal*.
[29]. Texas Instruments, *TMS320C28x DSP CPU and Instruction Set-Reference Guide* , Literature Number: SPRU430B, August 2001 – Revised May 2002.
[30]. Vojin Ž ivojnovic and Heinrich Meyr, *Compiled HW/SW co-simulation*, Proceedings of the 33rd annual conference on Design automation conference, pp.690-695, USA, June 03-07, 1996.
[31]. Chris Schlager, Joachim Fitzner and Vojin Zivojnovic, *Using Supersim Compiled Processor Models For Hardware, Software And System Design*, (Details not available)
[32]. A. Ghosh, M. Bershteyn, R.Casley, C. Chien,A. Jain, M. LIpsie, D. Tarrodaychik, and O. Yamamoto, *A Hardware-Software Co-simulator for Embedded System Design and Debugging*, In proceedings of ASP-DAC'95.
[33]. C. A. Valderrama, A. Changuel, P. V. Raghavan, M. Abid, T. B. Ismai and A. A. Jerraya, *Unified Model for Co-simulation and Co-synthesis of mixed hardware/software systems*, Proc. EDAC'95, Paris, France, February - March 1995.
[34]. Luis Sanchez Fernandez,Gernot Koch, Natividad Martinez Mardrid Maria Luisa Lopez Vallejo, Carlos Delgado Kloos and Wolfgang Rosenstiel, *Hardware-Software Prototyping from LOTOS*, Design Automation for Embedded Systems, Kluwer Academic Publishers, 1998.
[35]. Edward A Lee, *Overview of the Ptolemy Project*, Technical Memorandum UCB/ERL M01/11 March 6, 2001.
[36]. P.G. Plöger and J. Wilberg, *A Design Example using CASTLE Tools*, Workshop on Design Methodologies for Microelectronics, Institute of Computer Systems Slovak Academy of Sciences, Bratislava, Slovakia, pp. 160-167, Sep., 1995.
[37]. J. Wilberg, A. Kuth, R. Camposano, W. Rosenstiel and H. T. Vierhaus, *Design Space Exploration in CASTLE*, Workshop on High-Level Synthesis Algorithms, Tolls and Design (HILES), Stanford University, Nov. 1995, in: GMD-Studie Nr. 276, Dec. 1995.
[38]. Achim Osterling, Thomas Benner, Rolf Ernst, Dirk Herrmann, Thomas Scholz and Wei Ye, *The Cosyma System*, a chapter from Hardware/Software Codesign: Principles and Practice, pp 263-282, Kluwer Academic Publishers, 1997.
[39]. J. Madsen, J. Grode, P. V. Knudsen, M. E. Petersen and A. Haxthausen, *LYCOS: the Lyngby Co-Synthesis System. Design Automation of Embedded Systems*, Vol. 2, No. 2, March 1997.
[40]. Achim Osterling, Thomas Benner, Rolf Ernst, Dirk Herrmann, Thomas Scholz and Wei Ye, *The Lycos System*, a chapter from Hardware/Software Codesign: Principles and Practice, pp 283-305, Kluwer Academic Publishers,1997.
[41]. W. Fornaciari, D. Sciuto and A. Balboni, *Partitioning and Exploration Strategies in the TOSCA Co-Design Flow*,4th International Workshop on Hardware/Software Co-Design (Codes/CASHE'96),Pittsburgh, Pennsylvania , 1996.
[42]. R. K. Gupta and G. De Micheli, *A Co-Synthesis Approach to Embedded System Design Automation. Design Automation for Embedded Systems*, January 1996.
[43]. Rajesh Kumar Gupta,*Co-Synthesis Of Hardware And Software For Digital Embedded Systems*, Phd. Thesis, Dept. of Electrical Engineering, Stanford University, 1993.
[44]. *HardwareC-A language for Hardware Design*, (Details not available)
[45]. Pai H. Chou, Ross B. Ortega and Gaetano Borriello, *The chinook Hardware/Software Co-Synthesis System*, Proceedings of the eighth international symposium on System synthesis,France, 1995.
[46]. Tarek Ben Ismail and Ahmed Amine Jerraya, *Synthesis Steps and Design Models for Codesign*, Computer, Vol. 28, No. 2,pp. 44-52, February 1995.
[47]. Telelogic, *Specification and Description Language (SDL)*, (Details not available)
[48]. Verkest, K. Van Rompaey, Ivo Bolsens and Hugo De Man, *CoWare-A Design Environment for Heterogeneous Hardware/Software Systems*, Design Automations for Embedded Systems, 1(4), 357-386, 1996.

[49]. L. Lavagno, M. Chiodo, P. Giusto, H. Hsieh, S.Yee, A. Jurecska, and A. Sangiovanni-Vincentelli , *A Case Study in Computer-Aided Co-Design of Embedded Controllers*, In Proceedings of the International Workshop on Hardware-Software Co-design, pp. 220-224. 1994.
[50]. Gerard Berry, *The Esterel V5 Language Primer, Version 5.21 release 2.0*, April 6, 1999.
[51]. D. D. Gajski, F. Vahid, S. Narayan, and J. Gong,*SpecSyn: An Environment Supporting the Specify-Explore-Refine Paradigm for Hardware/Software System Design*, IEEE Transactions on VLSI Systems 6, no 1, pp. 84-100 1998.
[52]. D.D. Gajski, F. Vahid, S. Narayan, and J. Gong, *Specification and Design of Embedded Systems. Englewood Cliffs*, NJ: Prentice Hall, 1994.
[53]. S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, *Optimization by simulated annealing. Science*, 1983.
[54]. D.D. Gajski et al., *Specification Languages*, presentation slides, September 2000.
[55]. Andreas Mitschele-Thiele, *Systems Engineering with SDL*, Wiley, 2001.
[56]. B. Bolognesi and E. Brinksma, *Introduction to the ISO Specification Language LOTOS*, Computer Networks and ISDN Systems 14, pp. 684-707, 1987.
[57]. Stanislaw Budkowski, *Estelle: ISO-Formal Description Technique*, National Institute of Telecommunications, France, 1989.
[58]. G. Berry, *Hardware implementation of pure Esterel*, In Proceedings of the ACM Workshop on Formal Methods in VLSI Design, January 1991.
[59]. David Harel, *Statecharts: A visual formalism for complex systems*, Science of Computer Programming, 8:231-274, 1987.
[60]. Robert Esser, *An object oriented Petri net language for embedded system design*, In: Proceedings of the 8th International Workshop on Software Technology and Engineering Practice incorporating Computer Aided Software Engineering, London, 1997.
[61]. D. Smit, *VHDL & Verilog Compared & Contrasted*, Proc. 33rd Design Automation Conference, 1996.
[62]. Peter J. Ashenden, *Verliog and Other Standards*, IEEE Design & Test of Computers, pp. 84-85, January 2002.
[63]. Rainer Dömer, *The SpecC Language*, A Tutorial Presentation, Centre for Embedded Computer Systems, University of California, Irvine. (Date not available)
[64]. Andreas Gerstlauer, *The SpecC Methodology*, A Tutorial Presentation, Centre for Embedded Computer Systems, University of California, Irvine. (Date not available)
[65]. Rainer Dömer, *System-Level Modeling and Design with the SpecC Language*, PhD Thesis, University of Dortmund, 2000.
[66]. C.A.R Hoare, *Communicating Sequential Processes*, Prentice Hall International, First Publication 1985, March 2003.
[67]. Daniel C. Hyde, *Introduction to the Programming Language Occam*, Department of Computer Science, Bucknell University, Lewsiburg, Updated March 20, 1995.
[68]. P. Hilfinger, *A High-Level Language and Silicon Compiler for Digital Signal Processing*, In proceedings of the Custom Integrated Circuits Conference, -NA, 1985.
[69]. Stan Y. Liao, *Towards a New Standard for System-Level Design*, CODES'00, 2000.
[70]. Celoxica Limited, *Handel-C Reference Manual*, 2001.
[71]. Cedric Alquier, *Stephane Guerinneau, Lauro Rizzatti and Luc Burgun, Co-Simulation Between SystemC and a New Generation Emulator,* DesignCon 2003.
[72]. Xilinx Website, *www.xilinx.com*
[73]. Sanjiv Narayan, Frank Vahid, Daniel D. Gajski, *System Specification with the SpecCharts Language,Design & Test of computers*, pp 6-13 (Vol. 9, No. 4), October/December 1992.
[74]. A. Birrell and B. Nelson, *Implementing Remote Procedure Calls*, ACM Transactions on Computer Systems, 1984.

Integration of a Stereo Vision System and GPS Data for Recording the Position of Feature Points in a Fixed World Coordinate System

S.D. Mcloughlin
School of Informatics and Engineering
Institute of Technology Blanchardstown
Dublin 15
simon.mcloughlin@itb.ie
C. O'Rourke
Department of Computer Science
NUI Maynooth
Co. Kildare
colin.orourke@may.ie
J. McDonald
Department of Computer Science
NUI Maynooth
Co. Kildare
jmcd@cs.may.ie
C.E. Markham
Department of Computer Science
NUI Maynooth
Co. Kildare
charles.markham@may.ie

Abstract

This paper describes a laboratory system for recovering the global coordinates of feature points obtained from a moving camera. The prototype includes a stereo vision system combined with an overhead camera, which mimics a GPS receiver. The stereo vision system provides three dimensional feature point coordinates relative to the position of the cameras and the overhead camera provides three-dimensional coordinates of the camera in a "global" coordinate system. The fusion of these data provides three-dimensional feature point coordinates in a fixed origin global coordinate system.

Keywords: Stereo vision, GPS, Data Fusion

1 Introduction

Many applications currently exist that require the position of an arbitrary feature point in a fixed origin coordinate system. A mobile road integrity inspection system may need to log the position of a faulty or obsolete road marking or a pothole. An ordnance surveyor may need to record the location of an object for map reconstruction.

In this paper a system is described that addresses these problems. The system is divided into two main components. The first is a stereo vision system, which is capable of extracting local three dimensional coordinates of a point in the scene. Unfortunately, every time the cameras

move, so does the origin of the camera coordinate system so feature points do not have a fixed origin under camera motion.

The second component of the system introduces a fixed origin global coordinate system for feature points identified under arbitrary camera motions. Currently the system is a laboratory prototype consisting of an overhead camera extracting the position and orientation of a small remote controlled mobile stereo vision system. The final system will use a GPS receiver to obtain position information. The moving compass built into many GPS receivers or an inertial navigation sensor will provide orientation information. A transformation of the stereo coordinate systems to this world coordinate system results in the coordinates of feature points in a fixed origin world coordinate system.

Previous work has seen the development of similar systems for navigational purposes. Simon and Becker [1] fused a DGPS sensor and a stereo vision sensor to recover global position of a vehicle to examine road geometry. Rock et al [2] fused a CDGPS sensor and a stereo vision module to aid a helicopter in the tracking of an object. The purpose of the system described in this paper is to provide a mechanism for a road engineer to record global positions on a standard map of faulty road markings identified automatically in a prerecorded image sequence.

A more detailed description and explanation of the system components and techniques is presented in section 2. Some preliminary results obtained form the prototype is presented in Section 3. Section 4 concludes the paper and discusses some future work in the area.

2 System description

This section describes in detail the two components of the system along with their fusion. Section 2.1 describes the stereo component. Section 2.2 describes the navigational component and Section 2.3 describes the fusion of the two.

2.1 Computation of local three-dimensional coordinates of feature points

The apparatus used to compute local three-dimensional coordinates of feature points is shown in figure 2.1. It consists of two SONY XC-77CE CCD camera modules (576 x 768) mounted on a mobile platform which is controlled remotely. The remote control car on which the chassis was obtained was chosen for its slow speed and robust design, it being designed for

young children. Video and synchronization signals were connected to the car via a cable harness, (see Figure 1).

Figure 1 Mobile remote controlled stereo vision unit

The model provides a scaled down version of the vehicle-mounted system under development. The baseline of the camera pair is 0.0625. The baseline of the real system is approximately 1.2 meters to allow for ranging of objects a significant distance in front of the vehicle.

Conventional techniques for stereo ranging were used. The epipolar geometry of the cameras is computed from the fundamental matrix [3]. Correlation is conducted along the epipolar lines to facilitate stereo feature point correspondence [4].

2.2 Computation of vehicle coordinates and orientation in global coordinate system

A black enclosure was constructed (3 meters x 2 meters) to serve as a "world" space, (see Figure 2). An overhead camera was placed directly above the enclosure to extract the position and orientation of the mobile unit. A white arrow was placed on top of the mobile unit to facilitate this process. Position is computed as the center of mass of the arrow. Orientation of the arrow marked on the roof of the vehicle was computed using a technique based on the method of least squares [5], (see Figure 3). An alternative approach to identifying orientation would have been to use the Hough Transform.

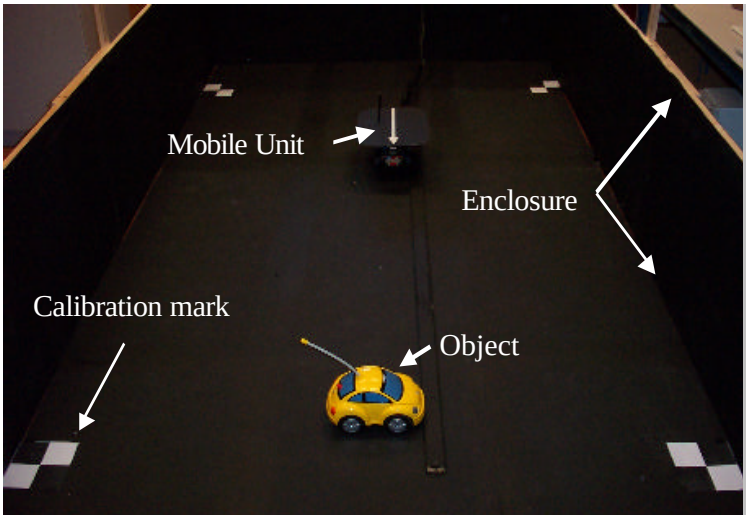


Figure 2 System configuration, showing location of key components

2.3 Fusion of local and global coordinate system

The local and the global coordinate systems are Euclidean. The transformation of the local coordinate axes to the global coordinate axes will give the location of the feature point in the global coordinate system. Let θ be the orientation of the mobile unit. Let **T,** be the position vector of the mobile unit. The rotation matrix for the x and z coordinates is,

$$\mathbf{R} = \begin{bmatrix} \cos\left(\theta - \frac{\pi}{2}\right) & -\sin\left(\theta - \frac{\pi}{2}\right) \\ \sin\left(\theta - \frac{\pi}{2}\right) & \cos\left(\theta - \frac{\pi}{2}\right) \end{bmatrix} \quad \textbf{(2.1)}$$

The factor of $-\pi/2$ was introduced since the direction of the arrow is perpendicular to the baseline direction. The location of the feature point in the global coordinate system is denoted as P_g. The location of the feature point in the local coordinate system is denoted as P_l. P_g can be computed from the transformation,

$$\mathbf{P}_g = \mathbf{P}_l\mathbf{R} + \mathbf{T} \quad \textbf{(2.2)}$$

The y coordinates (height information) of the feature point do not require transformation assuming the vehicle stays horizontal to the plane of the road.

3 Results

The mobile unit was guided under remote control along a predefined path. The frame grabber (SNAPPER 24 bit) captured simultaneously the stereo pairs and overhead frames. This was achieved by connecting each camera to the separate red, green and blue channels of the frame video capture card. A common synchronization pulse was provided by a fourth camera. The resulting image sequence was processed to provide position and orientation data of the mobile unit for each stereo pair, (see Figure 3).

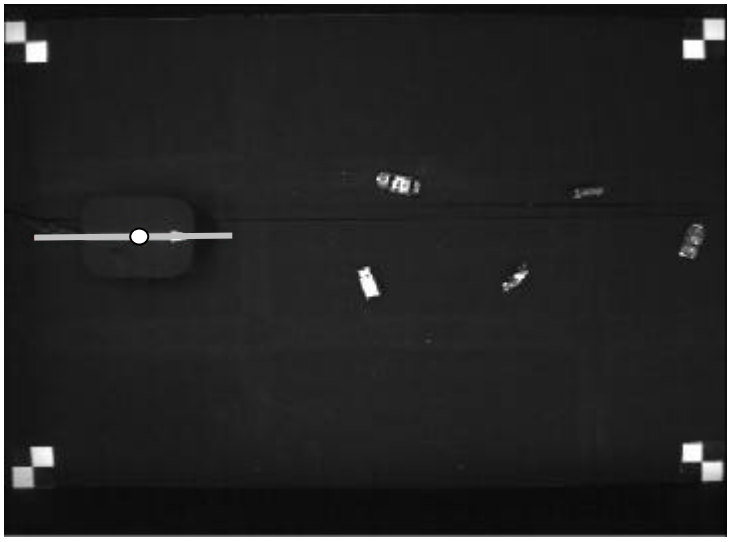
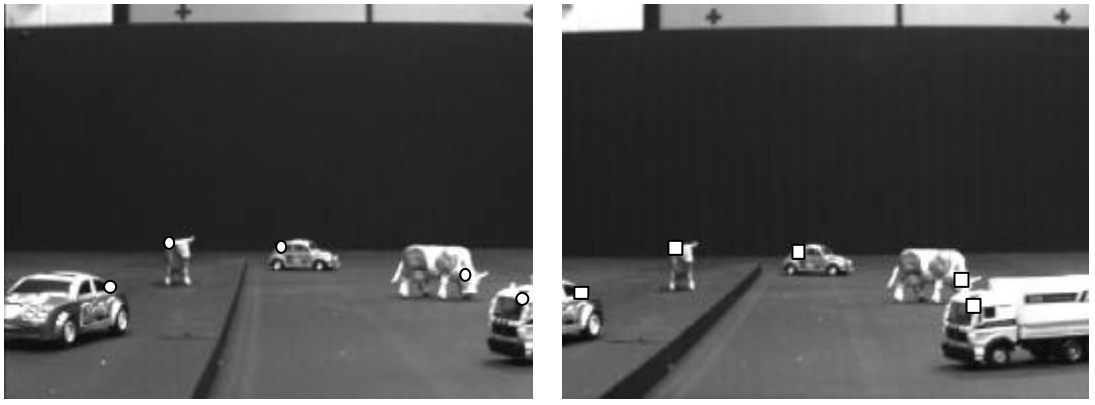


Figure 3 Overhead view of the scene.
Position and orientation of the mobile unit have been superimposed.

Feature points were identified in the stereo pair and the local coordinate evaluated (see Figure 4).



Left Image Right

Figure 4 Feature points identified in left and right images of the stereo pair

These data sets were then fused to produce global coordinates of the feature points. The feature points were then superimposed on an aerial view to provide a visual demonstration that the system was working (see Figure 5).

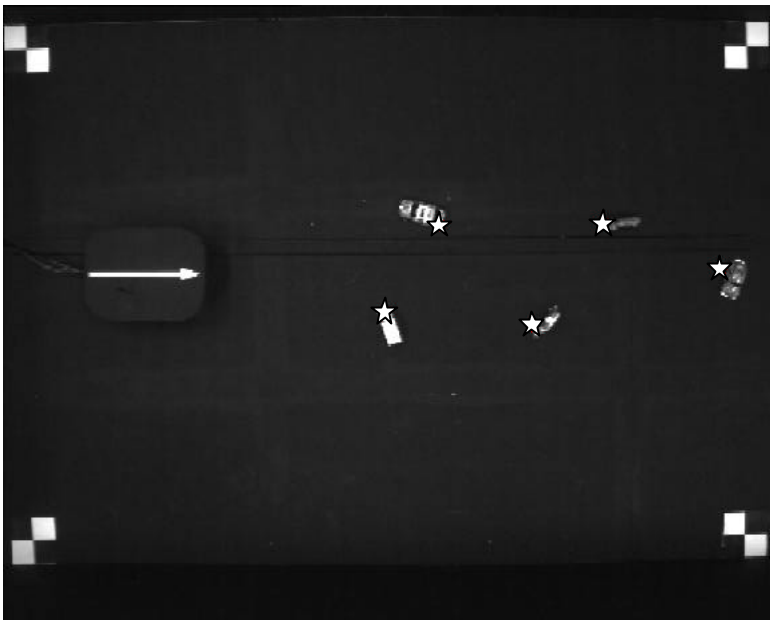


Figure 5 Overhead view of the scene.
Fused co-ordinates of feature points have been over plotted using stars.

4 Conclusion

The aim of this paper was to describe a method for fusing data provided by a mobile vision system with a GPS receiver. The results clearly demonstrate that such a system is realizable. The work is part of a larger project aimed at designing a system for automated inspection for infrastructure management. The software developed will form part of a full-scale system currently being developed. This system currently consists of a GPS receiver (Garmin 3+) and mobile stereo acquisition system with a 1.2m baseline. A further extension to this work will be to use the global position to assist in feature point tracking between frames.

References

[1] A. Simon, J. Becker, "Vehicle Guidance for an Autonomous Vehicle", *IEEE International conference on Intelligent Transportation Systems*, Tokyo 1999.

[2] S.M. Rock et al, "Combined CDGPS and Vision-Based Control of a Small Autonomous Helicopter", *Proceedings of the American Control Conference*, Philadelphia, PA, 1998.

[3] R. Hartley, A. Zisserman, "*Multiple View Geometry*".

[4] O.D. Faugeras, "*Three-dimensional Computer Vision*", 1993.

[5] R. Jain, et al, "*Machine Vision*", 1995.

A Secure Payment Protocol

Mark Cummins
Institute of Technology Blanchardstown
mark.cummins@itb.ie

Abstract

iBank is a business model and a set of protocols allowing customers to pay owners and retailers of information. This paper outlines this new electronic commerce mechanism specifically designed for micro-payments. The iBank scheme employs a proof of transaction but is unique in the way the payment commitment is distributed. The remainder of this paper explains in detail the full payment protocol that has developed; this includes all aspects of the protocol, including how the protocol handles disputes.

1 Background

The growth of the Internet and the increasing sophistication and availability of cryptographic tools have promised to bring commerce to new heights of efficiency and international breadth. Efficiency suggests a number of things, including minimised human involvement, improved distribution of goods and information, and more rapid processing of transactions. Electronic payment over the Internet has great potential in terms of banking because much of the transaction may be automated. Indeed some believe that electronic payment over the Internet will become an integral part of commercial culture, in the same way that this has happened for the automatic teller machines and credit cards. Unfortunately, due to the insecure nature of the Internet, the security of electronic payment is unsure. In general the current public opinion is that transactions over the Internet are extremely risky and until the much-awaited SET[38] protocol is widely deployed few banking corporations are willing to guarantee the safe transaction of funds over the Internet.

One of the main problems with electronic payment on the Internet is that the transaction occurs remotely. Unlike normal payment there is no physical proof of the transaction occurring. Cryptographic tools are required in payment protocols to provide this proof in the electronic medium. All payment protocols intended for practical use need to take into account their efficiency with regard to processing, communications and storage requirements. Cryptographic processing, particularly with asymmetric algorithms, is computationally expensive, so it is important to minimise the use of cryptography to where it is really required. The SET protocol has been criticised for its intensive use of asymmetric cryptography. Some

[38] Secure Electronic Transaction protocol developed by Visa and MasterCard for secure online credit card transactions

protocols, particularly those for micro-payments, are willing to sacrifice some security in return for large gains in efficiency. The iBank protocol proposed in this paper maintains high security, but is more efficient in terms of messages transmitted and computer processing required than other protocols with similar aims.

1.1 Introduction

As the explosive growth of the Internet continues, more people rely on networks for timely information. However, since most information on the Internet today is free, intellectual property owners have little incentive to make valuable information accessible through the network. There are many potential providers who could sell information on the Internet and many potential customers for that information. What is missing is an electronic commerce mechanism that links the merchants and the customers. iBank is a business model and a set of protocols allowing customers to pay owners and retailers of information.

The major advantage of the iBank scheme, in comparison with known schemes, is that it allows the merchant to act in a completely passive way. This makes the scheme ideal for use in an Internet environment where merchants already almost universally provide web sites where their customers can download product information. With the iBank scheme merchants can provide goods to be downloaded by the customers, but these goods remain encrypted until the customer has committed to payment. As a result the merchant performs no on-line processing. A major benefit of this is that network processing on the merchant side is completely unchanged from the current technology.

The iBank scheme employs a proof of transaction but is unique in the way the payment commitment is distributed. Specifically, in comparison with all the known schemes in the literature, the transmission of the proof of transaction is reversed; instead of being transferred from customer to merchant to bank, it is transmitted from merchant to customer to bank. The remainder of this paper explains in detail the full payment protocol that I have developed, this includes all aspects of the protocol including how the protocol handles disputes. I also outline further possible developments of the complete iBank model.

1.2 The market for information

Porat and others have shown that information industries dominate the economy [1]. Estimates of the market for on-line information vary from $20 billion to $180 billion per year depending upon how the market is defined [2]. Vendors can distribute information products varying from

complex software valued at thousands of Euro per copy, to journal pages or stock quotes valued at a few cents each. A challenge for network-based electronic commerce is to keep transaction costs to a small fraction of the cost of the item. The desire to support *mircopayments* worth only a few pence each is a driving factor in our iBank design.

A second challenge in the information marketplace is supporting *micromerchants*, who may be individuals who sell relatively small volumes of information. Merchants need a simple way of doing business with customers over networks, so that the costs of setting up accounting and billing procedures are minimal. The purchase of goods over a network requires linking two transfers: the transfer of the goods from the merchant to the customer, and the transfer of money from the customer to the merchant. In the case of physical goods, a customer can order the goods and transfer money over the network, but the goods cannot be delivered over the network. Information goods have the special characteristic that both the delivery of the goods and the transfer of money can be accomplished on the same network. This allows for optimisations in the design of an electronic commerce system.

1.3 Related work

The design of most electronic payment protocols exists within a common framework. This framework is traditionally based on credit card payment models and consists of three entities. The customer entity is the user and the entity, which wishes to make a purchase of the goods or services. The merchant entity is the shopkeeper who uses the payment scheme to offer goods and services to the customer entity. The bank entity is responsible for the transfer of funds between the other two entities.

The establishment and distribution of cryptographic keys and other data flows between the different entities in payment protocols exhibit the following common properties:

- A bit string containing certain information is used to convey value. The information varies depending on the type of payment protocol. This bit string is often called a coin or payment commitment in the literature, and here we call it the proof of transaction. Once transmitted it can be used as proof that an entity agrees or commits to the terms of the transaction.
- The proof of transaction is usually digitally signed to indicate who transmitted it or is encrypted in such a way that it is known as to whom sent the message.
- This proof of transaction is created by the bank and the customer for the merchant. During the payment process it is usually sent from the customer to the merchant. The

merchant sends it to the bank to verify its authenticity and thus verify the payment transaction.

The integrity of the proof of the transaction is used to prevent or discourage internal threats. The protection of the proof of transaction during transmission is important to defend against both external and internal threats. That is why the proof of transaction is almost always transmitted along authenticated channels.

The iBank scheme employs a proof of transaction but is unique in the way the payment commitment is distributed. Specifically, in comparison with all the known schemes in the literature, the transmission of the proof of transaction is reversed; instead of being transferred from customer to merchant to bank, it is transmitted from merchant to customer to bank. In the iBank scheme the merchant creates the payment commitment (called a voucher) with the assistance of the bank. The payment commitment is then distributed to potential customers. The customer has the option to complete the transaction by signing the bit string. The payment commitment is then sent to the bank to verify the validity of the transaction.

The major advantage of the iBank scheme, in comparison with known schemes, is that it allows the merchant to act in a completely passive way. This makes the scheme ideal for use in an Internet environment where merchants already almost universally provide web sites where their customers can download product information. With the iBank scheme merchants can provide goods to be downloaded by the customers, but these goods remain encrypted until the customer has committed to payment. As a result the merchant performs no on-line processing. A major benefit of this is that network processing on the merchant side is completely unchanged from the current technology.

It should be made clear that in comparison with previously proposed schemes, the reduction in merchant processing is accompanied by a consequent shift in processing to the bank. However, overall there is a saving from the widely distributed merchants to the central banking facility where it is natural to concentrate processing power, and where processing may be aggregated.

2 An iBank scenario

A user wishes to buy information from a merchant's server. An iBank server maintains accounts for both customers and merchants. These accounts are linked to conventional financial institutions. An iBank transaction transfers the information goods from merchant to

user, and debits the customer's account and credits the merchant's account for the value of the goods. When necessary, funds in a customer's iBank account can be replenished from a bank or credit card; similarly funds in a merchant's iBank account are made available by depositing them in the merchant's bank account.

The transfer of information goods consists of delivering bits to the customer. This bit sequence is the encrypted version of the information that the user requested; then the user pays for the decryption key and downloads this key from the iBank server. This decrypted bit sequence may have any internal structure, for example, a page of text, or a software program. Once the customer receives the bits, there are no technical means to absolutely control what the customer does with them. Similarly, there is no technical means to prevent users from violating copyright by redistributing information [3].

2.1 A Closer look at the iBank Model

In a real world voucher scheme, merchants create vouchers, which allow customers to receive discounts for the purchase of goods or even allow the customer to redeem them for the actual goods. This is the basis of the iBank scheme. This payment scheme is ideal for processing of electronic goods in a communications network, such as software or electronic news and information. However, it could be modified to work for physical goods as well by substituting an authorisation message or a release signal for the electronic goods package.

The transaction model has a traditional configuration consisting of merchant, bank and customer entities. In reality the bank entity often represents two separate parties; the acquirer and the issuer. The acquirer is the merchant's bank and the issuer is the customer's bank. If the acquirer and the issuer are separate entities we assume that they share a secure private communications link. This scheme consists of three parts. Each of these parts must be completed to successfully conduct a transaction.

1. The merchant and the bank work together to create a voucher, which contains the important transaction commitment. A major saving of the scheme is that this process is only required once for a particular item. Since electronic items are likely to be purchased many times, this is a significant advantage.
2. The merchant allows the free distribution of the voucher to customers. For electronic goods this would typically mean that the voucher is placed on the merchants web site. The voucher will be distributed together with the actual electronic goods. However,

the goods are encrypted so that during this phase the customer cannot access the goods.

3. The customer and the bank co-operate and enable the customer to decrypt the electronic goods and thus allow the customer to access the goods.

The iBank protocol is designed to achieve the following main objectives. These are common objectives which well-known schemes such as iKp[3] and NetBill[4] payment systems attempt to achieve:

1. The customer and the merchant are able to agree on the details of the transaction.
2. The customer can only receive the specified goods if she has paid for them.
3. The customer and merchant have proof that a successful transaction has occurred.
4. The customer is able to protect her identity from the merchant.
5. The transaction cannot be adversely affected by external threats. This means that a malicious party (which is not included as part of the payment model) cannot prevent the transaction from succeeding. This can either be by causing the merchant not to be paid or preventing the customer from receiving the goods.
6. The transaction is efficient in terms of processing and in the number of messages required to complete the protocol.

3 The iBank Protocol

The protocol description will use the notation i. X Y to indicate that the i^{th} message of the protocol is transmitted by entity X to entity Y. The customer entity is represented by C, the merchant entity is represented by M and the bank or notary is represented by B. The basic voucher protocol consists of five steps:

M B : Request key
B M : Return key

M C : Distribute voucher
C B : Redeem Voucher
B C : Release Goods

The horizontal line after the first two steps separates the protocol part that need be preformed only once, in an offline exchange between bank and merchant. The remaining steps may be run repeatedly with different customers at any time. Thus the merchant engages in no online processing.

- Objective 1 is achieved in steps 3 and 4 of the protocol. The merchant determines the price for his goods and sets this in the merchant signed voucher. The customer agrees to this price by cashing in the voucher.
- The customer can only decrypt the goods by using the secret key. The customer receives this after the bank has determined that the transaction should go ahead and the funds have been transferred from the customer's account. Thus objective 2 is realized.
- The goods decryption key received by the customer is her proof that the bank has successful conducted the transaction. The merchant's proof of transaction is the bank signed product identification code sent in step 5 of the protocol. Objective 3 has been achieved.
- The customer's identity is protected from the merchant because the customer does not transmit any messages directly to the merchant. Also even if the customer does have a voucher, she is under no obligation to cash in the voucher and decrypt the goods. Thus objective 4 is attained. Note that while anonymity from the merchant is obtained, complete anonymity for the customer is not obtained because the bank will be able to identity the customer and the goods that she purchases.
- Objective 5 is achieved because no message within the protocol contains data that is either unsigned or unencrypted.
- Objective 6 is discussed in later sections. But steps 1 and 2 of the protocol are performed offline and only need to be conducted occasionally and do not need to be conducted for every transaction. This decreases the number of messages required for an average transaction. The iBank payment scheme only requires 1 signature from each entity (the merchant does not even need to do this online) and a one way hash calculation from the bank and the merchant.

3.1 Notation

The following notation is used to denote cryptographic operations. X and Y always represent communicating parties. K always represents a cipher key. When describing the following protocols, the sequence of messages is exchanged among three parties: C, the customer, M, the merchant: and B the bank, acquirer or notary entity.

Exy(Message), Message, encrypted with the key XY using symmetric key cryptography. It is assumed that only X and Y knows the key, and that only these entities may know the contents of Message.

Sig_x(Message), Message, digitally signed by X using a suitable signature algorithm, such as RSA [12]. This implies that X's public key is used to ensure that the message was transmitted by X

H(Message), A cryptographic function which results in a digest and checksum of Message, using an algorithm such as the secure Hash algorithm (SHA) one-way hash function.

3.2 Voucher Creation Phase

This phase assumes that the merchant and the bank have exchanged RSA public keys so that they are able to verify the authenticity of digital signatures created by the other entity.

M B : Sig_M(MID, E_{MB}(Merchant Account Details))
B M : Sig_B(E_{MB}(K),Expiry)

In step 1 the merchant makes a request of the bank for a voucher key. This key will be used to provide the security required by the voucher. The merchant must provide his merchant identification as well as details of his account with the acquiring bank. The acquiring bank stores this information so that any funds, which are owed to the merchant by customers, may be deposited in that account. It is optional that this information be correctly authenticated with a digital signature. In essence this will allow anyone to sell goods and become a merchant provided that they have a unique merchant identification number. The merchant may choose this number. Merchant account details, like all other account details, must not be allowed to be sent across open networks in the clear.

After it has generated a symmetric key for the merchant, the bank encrypts it and sends it to the merchant with an expiry date, which indicates when this key is no longer valid. As this key is the main element in the protocol, which provides security for the transaction it is essential that it be encrypted, so only the merchant and the bank know it. The bank must sign the key and expiry to prevent the merchant from receiving a false key from an external party.

At this point the merchant can create a voucher for each product. The merchant voucher will contain the actual electronic goods that the customer is purchasing. First, the merchant generates the key, which will encrypt his goods in the following manner:

Kp = H(K, MID, PID, Value)

K is the key provided by the bank. This key is only known to the merchant and the bank. MID is the merchant identity, which the bank already knows. The product identity (PID) is used to indicate the product, which the voucher contains. The merchant can choose to have a different PID for each copy of the product he sells or he can choose to have one product identity for each product or even for a range of products. The PID and MID are used to provide a unique identifier for the voucher which cannot be tampered with. The cost of the product is also included in this key so that customers cannot adjust the value to be paid for the product. It is assumed that the one way hash function is such that the key K cannot be determined even if multiple valid values for Kp, MID, PID, and Value are known. Now the goods must be encrypted with the key, which has been generated by the merchant. The next section will describe the contents of the voucher in more detail.

If the merchant issues one product identity for a number of products or if the merchant has many copies of the same product with the same product identity the customer who has purchased the goods may freely distribute the key Kp to other customers. This would mean that a customer could then obtain certain goods from the merchant and release them using the key she has been given by a friend who has purchased the goods. Unfortunately this protocol cannot prevent this scenario, which is in essence the same as software piracy. However, only the legitimate customer has a receipt from the bank, which can be used to prove custody of a legal copy. This is the same situation as with purchase of electronic goods on physical media.

This phase of the payment system only has to be conducted once. The merchant can continue to use the key K provided by the bank to prepare vouchers for all of his products indefinitely. In practice it is advisable that the merchant request a new key at regular intervals to maintain the security of the voucher. This is the reason for the inclusion of an expiry date when the key is issued by the bank.

3.2 Voucher Distribution Phase

Vouchers can be freely distributed by the merchant with the associated encrypted goods. The additional data, which is included in the voucher, is shown below. It is not essential for the security of the payment scheme that the merchant sign the voucher but as this signature need only be constructed once when the voucher is created it does allow the customer to verify that the voucher and goods they have downloaded originate from the correct source.

3. M C : Sig_M (E_{kp} (goods), MID, PID, Description, Value, Expiry)

The voucher package includes the encrypted goods that the voucher will allow the customer to access. The voucher package also includes the merchant and product identities. These are required to uniquely identify the product that the customer is purchasing. Also included is a human readable description of the product so the customer has some indication of the type of the product they are purchasing. The value of the product must also be included so the customer can determine whether the cost of the product is worthwhile. The expiry date is also included for the customer as an indicator of how long the voucher will be valid. The customer may download the voucher and decide not to purchase the goods without any loss. Because the merchant signs the voucher, the customer can be sure that the goods and the voucher contents have been received correctly providing, of course, that the merchant is not cheating.

3.3 Redeeming Vouchers

Now that the customer has obtained the encrypted goods and the voucher from the merchant and she has decided to purchase the goods, they must request the key from the bank to release the goods. Again it is assumed that the customer is able to establish a secure connection with the bank.

C B : Sig_C (MID, PID, Value, E_{CB} (Customer Account details), Counter)
B C : E_{CB}(Kp)

In step 4 of the protocol the customer sends to the bank, the merchant and product id's and the value of the product as well as details of their account and a counter value. The merchant and product identity and the value of the goods are obtained from the voucher the customer has received. The counter is a value that must be maintained by the bank and the customer. The purpose of the counter is to uniquely identify this message and prevent an external entity from replaying the message and thus draining of a customer's account of funds. A timestamp could be used instead of a counter provided that problems associated with synchronization are properly addressed.

It is only when the customer signs this message that the voucher is given value. Up till this stage the customer can abort the transaction. If the customer chooses to accumulate vouchers over a period of time, the customer may concatenate multiple sets of merchant identity, product identity and value bit strings and sign them all at once. This will reduce the amount of processing required by the customer.

The bank now makes a decision as to whether the transaction should take place. The bank must consider things like the availability of the customer's funds, and the trustworthiness of both the merchant and the customer and check that the counter value is valid. If the bank decides that the transaction should occur, the bank moves the correct amount of funds from the customers account to the merchant's account using the details provided by the customer and the merchant during the transaction. At this point the bank may also deduct any transaction, handling or other fees.

As the bank already knows the merchant's key K, the bank uses these additional values to calculate the key Kp. In step 5 of the protocol the bank returns the key Kp to the customer. When the customer obtains the key Kp they are able to decrypt the goods they have received with the voucher and complete the transaction.

After the funds transfer has occurred the bank has the option of notifying the merchant that the transaction took place. This notification is only to assist the merchant in updating his inventory and may not be essential for online software goods. When, or if, this notification occurs can be determined by agreement between the bank and merchant. Large value transactions could be batched and sent at the end of each working day. Unfortunately if the merchant chooses not to be notified by the bank he has no indication that a transaction has occurred.

3.4 Disputes

If the merchant or the customer is not satisfied that the transaction has been conducted successfully a dispute has occurred. The voucher payment scheme has a process which is able to deal with most disputes. It is assumed that both the customer and the merchant can trust the bank to be fair in all decisions.

The voucher dispute resolution protocol consists of the following step:

1. C B : Sig_C (Kp, Sig_M (E_{Kp} (Goods), MIP, PID, Description, Value, Expiry))

The message consists of the key Kp that the customer received from the bank in step 5 of the payment protocol. The remainder of the message is the merchant signed voucher the customer received in step 3 of the payment protocol. Because the voucher is signed by the merchant, the customer is unable to alter the contents of the voucher without detection. The re-transmission of the voucher, including the goods, in the dispute protocol will increase the

amount of traffic on the network but it is expected that the dispute protocol will not be required very often. The resolution of the dispute need not necessarily be referred to the bank. Any trusted third party may be used as a judge, provided that party has access to the transaction key K. The following sections describe how the judge may deal with potential complaints.

Incorrect Key In the case of this dispute, the customer claims that, the key that they received from the bank does not correctly decrypt the goods which were received in the voucher.When the judge receives the message from the dispute protocol he calculates the disputed key Kd using the MID, PID and Value fields from the voucher and the key K which is already known to him.

$$K_D = H(K, MID, PID, Value)$$

The judge then compares K_D and the key Kp received from the customer. If they match then the customer has received a legitimate key and the transaction should be rolled back. If the keys do not match, either Kp was altered by the customer or the customer has transmitted an incorrect Kp as part of the dispute protocol. In this case the transaction is not altered. It is assumed that the transmission of each message in the protocol occurs successfully and that the contents of each message is not altered by any network interference.If the customer is not satisfied with this result it is possible that the incorrect goods have been delivered in the voucher.

Incorrect Goods The customer may not be satisfied that the goods that she received are the goods that she has purchased. It could be that the merchant has incorrectly constructed the voucher, or that the merchant has encrypted goods that do not match the goods description included in the voucher. To check the goods, the judge verifies that the merchant has constructed the voucher correctly by calculating the key K_D and checking the correct key K_P as described in the previous section. The judge then decrypts the goods using Kp. A human arbitrator determines if the decrypted goods match the description provided in the voucher. If the goods cannot be decrypted, the merchant has incorrectly constructed the voucher by providing an incorrect merchant or product identity or key K. In both of these cases the transaction is rolled back and the money returned to the customer.

Incorrect Payment Amounts This type of dispute includes any disagreement on the amount charged for the goods. This includes both the possibility that the customer has been charged too much or the merchant has been paid too little. In the voucher payment scheme the value

that the merchant assigns to the product cannot be maliciously altered by the customer because the value is part of the key which is required to decrypt the electronic goods. Both the customer and the merchant indicate their agreement to the value to be paid for the goods by transmitting the value field correctly. The merchant indicates his requested goods value within the signed voucher and the customer indicates their agreement to that value by signing it and sending it to the bank in return for the key Kp.

4 Business model

As stated earlier, the iBank model is a secure payment protocol, but also a business model, which adds certain factors and properties to any implementation of a prototype system. Some of the challenges in making electronic commerce systems feasible include, account management and administration: Users and merchants must be able to establish and monitor their accounts. For iBank, user account administration is provided through WWW forms or the user desktop application. Using a standard WWW browser, an authorized user can view and change an iBank account profile, authorize funds transfer into that account, or view a current statement of transactions on that account.

iBank also does not rely on customers having to store secret keys and hence does not limit customers to any single machine. Any customer can access the service from any PC that has the iBank application. Automating account establishment for both customers and merchants is important for limiting costs. (Account creation is one of the largest costs associated with traditional credit card and bank accounts.) To begin the process, a customer retrieves, perhaps by anonymous FTP, a digitally signed iBank security module that will work with the user's WWW browser. Once the customer checks the validity of the security module, they put the module in place. They then fill out a WWW form, including appropriate credit card or bank account information to fund the account, and submit it for processing. The security module encrypts this information to protect it from being observed in transit. The iBank server must verify that this credit card or banking account number is valid and that the user has the right to access it. There are a variety of techniques for this verification: for example, customers may telephone an automated attendant system and provide a PIN associated with the credit card or bank account to obtain a password.

Because both customers and merchants maintain iBank accounts, inter-institutional clearing costs are not incurred for every transaction. iBank accounts provide a low cost mechanism to aggregate small value transactions before invoking a relatively high fixed cost conventional

transaction mechanism. Customers move money into their iBank account in large chunks (for example, €50 - €100) by charging a credit card or through an ACH (automated clearing houses) transaction. Similarly, money moves from a merchant's iBank account to the merchant's high street bank through an ACH deposit transaction.

5 Further Work

As described above, iBank is well suited for supporting commerce in information goods. However, the iBank model can also be extended in a variety of ways to support other types of purchases. For example, iBank could be used equally well for conventional bill paying. A customer could view a bill presented as a Web page; instead of buying information goods, we can think of the customer as buying a receipt for having paid the bill. Also we have not looked at the purchasing of physical goods, this could be done by allowing the user to purchase valid paid receipts for the goods. If the product to be bought is a one hour movie, it is likely that the customer will want to stream the data directly to a viewer, which conflicts with iBank's model of certified delivery. Alternative approaches such as using the standard iBank protocol to periodically buy a key for the next N minutes of an encrypted video stream could also be explored. Separately, I mentioned earlier how the prevention of illegal copying of information goods is impossible, however one could research means of embedding a unique watermark in each copy sold which would allow illegal copies to be traced to the source.

6 Summary

I have designed a new secure payment protocol, optimised for information goods and network services, which is extendable to physical goods. I have achieved my main objectives, by providing a secure, efficient payment system supporting micropayments and micromerchants. A prototype implementation using the protocol could be designed for ease of use and provide a reliable, easy to use application for both customers and merchants alike. This implementation would allow merchants and customers to access their account information and use the iBank system to purchase goods from any location, where the iBank software exists. This is a major benefit over other protocols, which rely on certificates and secret keys, which confine users to a single machine or access to a smart card reader.

7 References

[1] Porat, M., The Information Economy (US. Office of telecommunications, 1977)
[2] New York Times, June 7, 1992
[3] Jeff Hostetler, "*A Framework for Security*," 2nd WWW Conference, Chicago, Illinois,

October, 1994.

[4] Chaum, D., "Achieving electronic privacy", *Scientific American*, 267, No. 2, pp. 76-81, 1992

[5] Mihir Bellare, Juan A Garay, Ralf Hauser, Amir Herzberg, Michael Stelner, Gene Taudik and Michael Waidner. IKP, *A family of secure electronic Payment protocols*. In proceedings of the first Usenix Workshop on Electronic commerce.

[6] Benjamin Cox, J. D Tygar, and Marvin Sirbu. *Netbill Security and transaction Protocol.* In proceedings of the first Usenix Workshop on electronic commerce. New York, July 1996.

[7] Stefan Brands. *Electronic Cash on the Internet.* In proceedings of the Internet society 1995 Symposium on Network an Distributed System Security, pages 64-84, 1995

[8] Visa / MasterCard International. *Secure Electronic Transaction (SET) Specification book 2, formal protocol definition*. http://www.visa.com/cgibin/vee/sf/standard.html

[9] Gennady Medvinsky and B. Clifford Nenman. *Netcash: A design for practical Electronic Currency on the Internat.* In Proceedings of first ACM Conference on computer and communication security. Pages 162 – 196. ACM Press 1993.

[10] Ronald L. Rivest and Adi Shamir. *PayWord and mircoMint. Two simple micropayment Schemes*. http://theory.lcs.mit.edu/rivest/RivestShamir-mpay.ps

[11] Peter Wayner, *Digital Cash, Commerce on the net,* AP Professional publishers. ISBN 0-12-738763-3

[12] R. Rivest and L. Adleman. *A method for obtaining Digital signatures and public key cryptosystems*. Communications of the ACM, pages 120-126 1978

[13] Shingo Miyazaki, Kouichi Sakurai, *An Efficient Fair off-line Electronic Cash System with Partially Blind Signatures Based on the discrete Logarithm Problem, Financial Cryptography*, Lecture notes in Computer science 1465, Rafael Hirschgfeld (Ed.) 1998

http://www.itb.ie