Proceedings of the

Second York Doctoral Symposium on Computing

Heslington Hall, University of York, UK
30th October 2008

Editors:
Pierre Andrews
Thomas Lampert

Scope of the Symposium

The aim of the *2nd York Doctoral Symposium on Computing* (YDS) is to establish a platform for dissemination and exchange of up-to-date scientific information on theoretical, generic and applied areas of computing by giving doctoral students an opportunity for presenting their ongoing research in computer science and scientific computing. In particular, the symposium intends to bring together young researchers who are active in this wide field and interested in an interdisciplinary exchange of ideas and experience. The symposium also strives to promote research and development for the improvement of interdisciplinary application of computing.

Topics include, but are not restricted to:

– Artificial Intelligence
– Bioinformatics and Mathematics
– Communications
– Database Systems
– Distributed Systems
– High Integrity System Engineering
– Human Computer Interaction
– Management and Information Systems
– IT Security
– Natural Language Processing
– Non-standard Computation
– Programming languages and systems
– Real-time systems
– Signal Processing and Pattern Recognition
– Theoretical and Computational Chemistry

Organization

YDS'2008 is organised and funded by the Department of Computer Science, Univeristy of York, UK.

Executive Committee

Conference Chair:	Maria Arinbjarnar (University of York)
Programme Chair:	Thomas Lampert (University of York)

Organising Committee

Committee Members:	Maria Arinbjarnar (University of York)
	Rania Hodhod (University of York)
	Thomas Lampert (University of York)
	Sofianiza Abd Malik (University of York)
	Jan Tobias Mühlberg (University of York)
	Richard Ribeiro (University of York)
	Marcelo Romero (University of York)
	Gul e Saman (University of York)
	Malihe Tabatabaie (University of York)

Thanks

Our thanks go to Jim Woodcock for the idea of organising YDS as well as for many stimulating comments and inspirations during the last couple of months. Furthermore, we thank Heather Barber for her language corrections and the departmental secretary Mandy Kenyon who was extremely helpful in organising this event.

Programme Committee

Programme Chair:	Thomas Lampert (University of York)
Programme Co-Chair:	Gul e Saman (University of York)
Committee Members:	Maria Arinbjarnar (University of York)
	Neil Audsley (University of York)
	Jim Austin (University of York)
	Alan Frisch (University of York)
	Edwin Hancock (University of York)
	Rania Hodhod (University of York)
	Chris Kimble (University of York)
	Daniel Kudenko (University of York)
	Sofianiza Abd Malik (University of York)
	Suresh Manandhar (University of York)
	Nick Pears (University of York)
	Helen Petrie (University of York)
	Marcelo Romero (University of York)
	Colin Runciman (University of York)
	Susan Stepney (University of York)
	Jan Tobias Mühlberg (University of York)
	Jim Woodcock (University of York)
	Richard Wilson (University of York)

Referees

David Andrews (University of Arkansas)
Pierre Andrews (University of York)
Jim Austin (University of York)
James Cussens (University of York)
Xiaocheng Ge (University of York)
Manuel López-Ibáñez (Napier University)
Simon O'Keefe (University of York)
Fiona Polack (University of York)
Richard Ribeiro (University of York)
Malihe Tabatabaie (University of York)
Philip Watson (University of York)

Table of Contents

I Invited Talks

II Session 1: Management and Information Systems

III Session 2: Architectures and Models

Part I

Invited Talks

Invited Talk: Building Appearance Models Using Groupwise Registration

David Cristinacce

School of Medicine, The University of Manchester
david.cristinacce@manchester.ac.uk

Abstract. The talk will give an overview of combined shape and texture models, which are able to model a population of a given class of objects which have distinctive landarks, for example human faces or medical images of a given organ.
Once constructed, appearance models can be used to rapidly fit to new examples of the same type of object. For example using the well known Active Appearance Model fitting algorithm and its derivatives. Alternatively the model can be used to analyse the statistical variations across a population.
However the initial construction of such models requires corresponding landmarks to be found across all examples of the training set. These landmark points are usually placed manually. However this approach is tedious and error prone. I will describe a method of finding point correspondences automatically using groupwise image registration, which enables model construction without manual intervention.
Examples of models built from human faces and from magnetic resonance imaging (MRI) brain images will be shown along with some discussion on the important factors when performing groupwise registration on different data sets.

Invited Talk: Learning in Computer Vision: Some Thoughts

Maria Petrou

Department of Electrical and Electronic Engineering, Imperial College
maria.petrou@imperial.ac.uk

Abstract. It will be argued that the ability to generalise is the most important characteristic of learning and that generalisation may be achieved only if pattern recognition systems learn the rules of meta-knowledge rather than the labels of objects. A structure, called "tower of knowledge", according to which knowledge may be organised, will be presented. A scheme of interpreting scenes using the tower of knowledge and aspects of utility theory will also be proposed. Finally, it will be argued that globally consistent solutions of labellings are neither possible, nor desirable for an artificial cognitive system.

Invited Talk: The Reduceron: Widening the von Neumann Bottleneck for Graph Reduction using an FPGA

Matthew Naylor

Department of Computer Science, The University of York
mfn@cs.york.ac.uk

Abstract. Reduction machines for functional languages will be explained, and a new one called the Reduceron presented. The Reduceron exploits wide, parallel memories to increase evaluation speed, and has been prototyped in programmable hardware. Results of a comparison with conventional machines, both in theory and in practice, will be given.

Part II

Management and Information Systems

Multi-Agent Based Peer-to-Peer Workflow Management System

A. Aldeeb, K. Crockett and M. J. Stanton

Department of Computing and Mathematics,
Manchester Metropolitan University, Manchester, M1 5GD, UK
{a.aldeeb, k.crockett, m.stanton}@mmu.ac.uk

Abstract. Current workflow management systems are under dynamic evolution and tend to move to distributed business processes. Software agents and Peer-to-Peer (P2P) technology are being recognized as a new approach to decentralized workflow management systems. Plugging Multi-agents into P2P workflow management systems can support run-time functions, workflow instance monitoring, adaptability and exception handling. In this paper, a novel Multi-agent based P2P architecture for a workflow management system is presented. The proposed system can be applied for an inter-organizational workflow scenario to form a virtual organization. The adoption of multi-agent within P2P network facilitates the capability of workflow actors to share, collaborate, and communicate in their own private web.

Keywords: Workflow, Workflow Management System, Multi-agents, Peer-to-Peer, Exception Handling, Virtual Organization

1 Introduction

Workflow is the computerized facilitation or automation of a business process in whole or part [22]. A workflow management system (WFMS) is a system that completely defines, manages and executes "workflows" through the execution of software whose order of execution is driven by a computer representation of the workflow logic [22]. Conventional workflow management systems are based on centralized client/server architecture. This requires a centralized database to store the workflow process definition and a centralized workflow engine to manage activities such as coordination and monitoring process execution [3, 4]. The main disadvantages of any such architecture are the potential bottleneck that can arise during process execution, and that the central database can become single point of fault.

Workflow processes are moving from long-lasting, well-defined, centralized business processes to dynamically changing, distributed business processes with many variants [3, 4, 13, 16]. Workflow applications are inherently distributed. They involve people, resources and tools that may be distributed over a wide geographic area [16]. Recent research has shown an increased interest in P2P based workflow systems [1, 2, 4, 8] to decentralize workflow systems. P2P is a way to develop a distributed applications at different nodes (Peers) to share resources and these nodes have symmetric roles either server or client. The P2P WFMSs are proposed to avoid the bottleneck and the central point of fault caused by centralized client/server workflow systems. P2P based workflow can also be used to improve scalability, system openness and support incompletely specified processes [4]. In P2P based WFMS peers join “virtual communities” according to their capabilities and discover each other using the services provided by an open P2P network. The coordination between peers is performed by exchange of notification messages.

In recent years, the integration of workflow and agent technology has attracted the attention of many researchers to support distributed business process in a dynamic and unpredictable environment [5, 6, 7, 14, 17, 18, 19, and 20]. Agents are persistent active entities that have the properties of autonomy, reactivity, and pro-activity and can perceive, reason, and communicate with other agents [10]. In addition, agents have the capability to dynamically form social structures through which they share commitments to the common goal of workflow enactment by forming a collective entity called Multi-agent systems [6]. In order to form and participate in multi-agent systems, they must be able to compromise on their autonomy level so they can coordinate with others [10]. Multi-agent can help in conducting the run-time function of WFMS, monitoring, adaptability and exception handling [19, 20].

The Multi-agent paradigm can be superimposed on the P2P architecture, where agents can reside in different peers and perform workflow tasks utilizing their decision support capabilities, the collective behavior, and the interaction protocols of the P2P system. Furthermore, P2P infrastructure supports dynamic service construction, modification and movement, and allows a dynamic agent to participate in multiple applications and dynamically form partnerships [12]. The contribution of this research can be summarized by the following statement: P2P system + Multi-agent = Adaptive Workflow Engine. The adoption of multi-agent within P2P network facilitates the capability of workflow actors to share, collaborate, and communicate in their own private web. The rest of this paper is organized as follow: section 2 describes the proposed Multi-agent P2P Architecture for WFMS. Section 3 outlines the adaptability and exception handling mechanism in the systems while section 4 presents a virtual organization application for the system. A case study and prototype implementation is presented in section 5 and section 6 concludes and states the future work.

2 Multi-Agent P2P Architecture for WFMS

A high level architecture of a Multi-agent based P2P WFMS is shown in Figure 1. This system is upgraded from previously designed P2P WFMS [1, 2] and identifies the major components and interfaces based on the Workflow Reference Model which released by the Workflow Management Coalition in October 1994 [22]. The new multi-agent version of the systems is proposed to take the run-time function and exception handling and workflow monitoring out of the P2P system level to multi-agent level. The build-time function is conducted at P2P level which includes workflow process modeling, storing process definitions and distributing the process to workflow agents deployed to the peers. The run-time function (Multi-agent level)

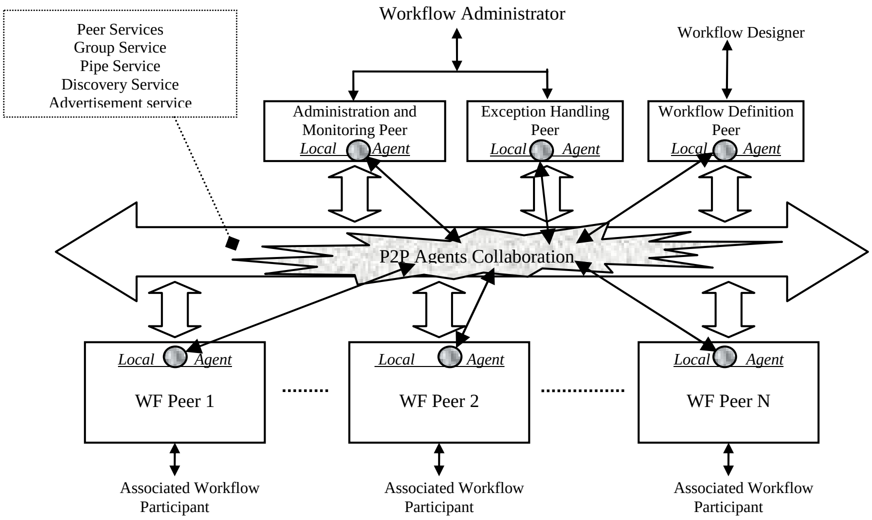

Fig. 1. Multi-Agent Based P2P WFMS

includes workflow process instantiation, task coordination and exception handling procedures. As shown in figure 1, the P2P network provides services that include advertisement services, group services, peer services, pipe services, and discovery services. In addition, it facilitates a user interface with human workflow participant. The workflow peers can be classified into three types based on their capabilities: Workflow Peer (WFP), Workflow Definition Peer (WFDP) and Exception Handling Peer (EHP). The WFP can reside on any machine on the P2P network enabling direct communication with other workflow peers to enact the workflow process. Each WFP is associated with a workflow participant and each uses a Local Agent (LA) to perform a part of the workflow. Once the task is completed, the LA informs its successor (Remote Agent – RA) at another node and the next task of the process may be executed. Process co-ordination is achieved by the exchange of messages between P2P agents. The second type of peers is WFDP which facilitates the design and the storage of the whole workflow schema at build-time. The workflow process is partitioned to separate tasks according to the roles of the workflow participants and the organizational structure. Then, the WFDP creates and deploys P2P agents loaded with their tasks to the corresponding peers. When a workflow agent is deployed, it registers its symbolic name and address at its destination node and keeps address book for other agents. A workflow process definition resulting from an evolutionary change is called a "version" of the workflow process. A workflow process definition resulting from an ad-hoc change which affects the running workflow instances is called variant of the workflow process [16]. P2P agents are able to learn and help in the workflow process evolution and play a key role in versioning mechanism. The third type of peers is the EHP which has the capability to deal with various types of exceptions. The exception handling procedure is discussed in the next section.

3 The Adaptability in Multi-Agent Based P2P WFMS

Two workflow exceptions can be identified in the proposed Multi-agent based P2P WFMS: local workflow exception and global workflow exception. Local workflow exception affects the task of local agent of one node. The local agent can handle this exception by applying one of two possible self-recovery policies; forward recovery or backward recovery. Forward recovery policy is based on correcting and isolating the effect of the exception and returning the workflow task to a normal state so the normal operation can be continued. In contrast, backward recovery policy is based on restoring the workflow task to a consistent state that occurred before the appearance of the exception. If the local workflow exception can not be handled within the affected local agent, it can propagate to the other agents in the other nodes leading to a global workflow exception. These types of exceptions will, of course, affect more than one node and a coordinating node is required to deal with this exception. In this research, the coordinating node which has the capability to deal with the exceptions is the exception handling peer (EHP). This peer uses a mobile agent to transfer to other nodes to captures exceptions, characterizes the exceptions and applies a recovery policy if these exceptions can not be handled locally at those nodes. Using the mobile agent leads to the reduction of the amount of the communication between nodes as the interaction will take place locally at the exception raising node [20]. In addition, The EHP will acquire some knowledge from previous exceptions and build an exception handling knowledge base. The Mobile Exception Handling Agent will be updated with this knowledge so it can respond to new similar exceptions rapidly by using exceptions knowledge learnt before. Using the EHP can guarantee the separation of business logic from exception handling logic. Mixing business logic and exception handling logic makes it difficult to keep track of both, complicating the verification of processes, as well as later modification [9].

In distributed systems, backward recovery of one process in a group of communicating processes will often require other processes in the group to be rolled back because of the interdependencies caused by message communication. The result is a cascade of rollbacks called the ‘domino effect’ [11]. To avoid the domino effect in the proposed system a conversation scheme is used. The conversation between P2P agents is formed by a group of P2P agents affected by an exception, and a workflow agent in the P2P agents conversation can only communicate with workflow peers that are in the same conversation. This can prevent the error propagation and limits the domino effect. The P2P agents’ conversation represents an atomic action consisting of interactions in a group of agents. After the effect of the exception is contained and resolved, the P2P agents’ conversation will be terminated and the workflow agents will return to the normal mode of the system.

4 P2P Agents Virtual Organization

In an Inter-Organizational workflow, the organizations involved have different legal and organizational systems, different security aspects, and heterogeneous hardware, operating systems and workflow applications. Inter-organizational workflow requires

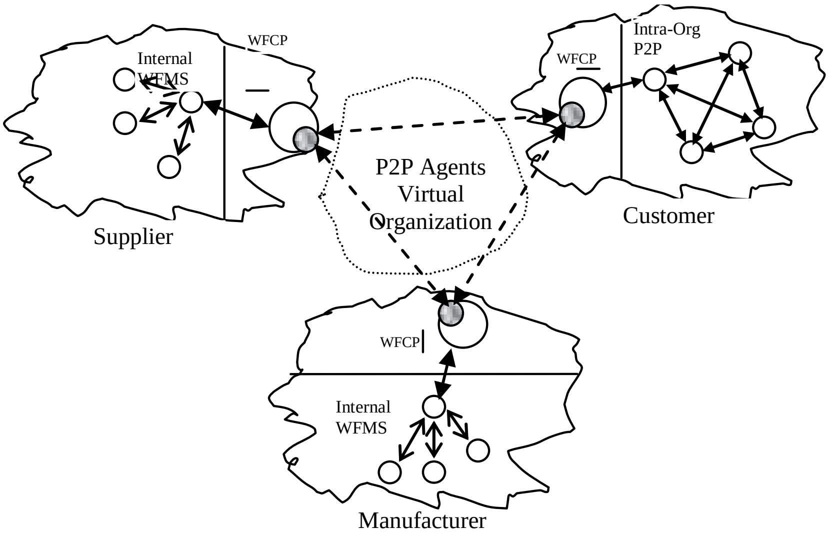

Fig. 2. P2P Agents Virtual Organization

flexible, on-the-fly alignment of business partners; in other words, adaptive workflow capabilities [6]. In inter-organizational WFMS, no central control exists and the partners are autonomous workflow actors, who can join and leave a virtual organization (VO) at any time. This scenario makes the proposed multi-agent based P2P WFMS a potential workflow solution. The organizations which involved in the inter-organizational workflow will be represented by workflow peers with workflow agent engines and these peers will form a Multi-agent based P2P VO. A public workflow model is agreed between different organizations which collaborate as peers, while keeping their internal private workflow within their boundaries.

Figure 2 shows an overview of the proposed Multi-agent based P2P inter-organizational WFMS working environment where three organizations customer, supplier and manufacturer are involved in managing of a workflow process. This includes; workflow advertisement, workflow interconnection, and workflow cooperation. Each organization acts as a workflow peer (WFP) in this process. The three workflow peers discover each other by the advertisement service provided by the P2P network infrastructure and the process enactment is conducted by multi-agent workflow engines. There are three phases of interactions between workflow peers.

The first phase is workflow peers identification where workflow peers for customer, supplier, and manufacturer publish their services and join different groups of a specific product or service e.g. (customers will form a group of customer of specific product). Each WFP needs to find a desired partner. The second phase is the multi-agent interconnection; starting with electronic negotiation and connection whereby a multi-agent based P2P virtual organization is formed. The third phase consists of multi-agent cooperation, instantiating a workflow cases and coordination of their tasks.

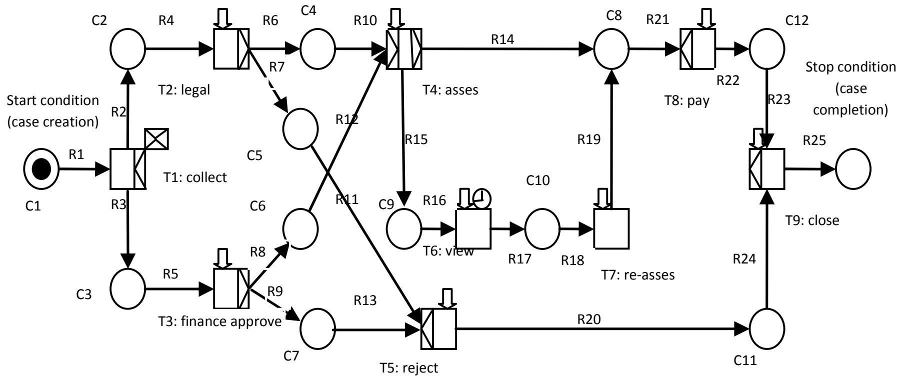

Fig. 3. Example insurance claim process in Petri net presentation

5 A Case Study and Prototype Implementation

To better illustrate how the proposed Multi-agent based P2P WFMS works, an example of a motor insurance claim process is presented. The process consists of nine tasks as shown in Figure 3. These tasks are distributed over the workflow agent associated with nodes and their workflow participants, based on the roles of the participants themselves and structure of the organization. To examine the system, both build-time and run-time functions are implemented. Build-time function implementation includes:

(1) Modeling the process using Petri Net (PN) notation (Figure 3). PN is chosen because it facilitates formal semantics despite the graphical nature and state-based instead of event-based [15]. Multi—agent can be easily modeled using PN notation, each TRANSITION in the PN diagram represents a workflow agent task while the PLACE corresponding to the pre-condition for that task. The dependency between any two nodes is represented by ARC. TOKEN is corresponding to the workflow instance.

(2) Transferring the Petri net model to a workflow schema stored in data base. The database design includes creating several tables, e. g a Workflow table to holds the workflow definition and its versions, PLACE table holds the details of each PLACE, TRANSITION table holds details of each TRANSITION and ARC table handles details for each ARC connecting different agents' tasks and their pre-conditions within the workflow process.

(3) Partitioning the stored process into tasks according to the roles of the workflow participants and the organizational structure of the insurance company.

(4) Creation of Workflow Agents based on workflow partitions.

(5) Deploying the agents to the relevant Workflow Peers using the P2P network.

After the build-time functions are completed, the system can carry out its run-time functions. These will include: first, instantiating a workflow case and coordination of

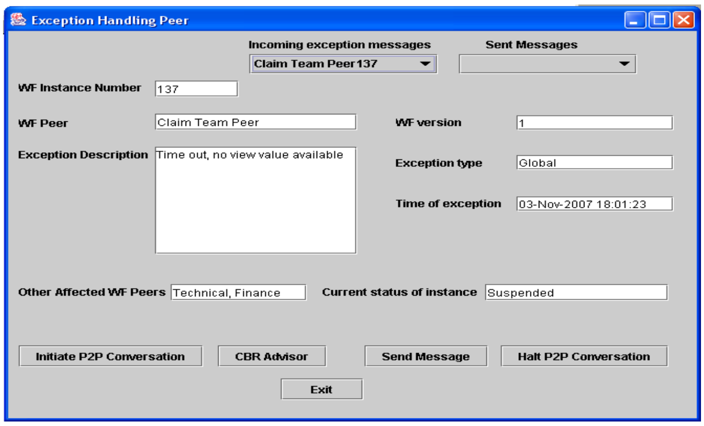

Fig. 4. Exception Handling Peer User Interface.

tasks between agents. Second, ad-hoc changes will be made to the running instances to examine the exception handling procedures. Exception handling can be achieved by backward recovery when the case attributes and all valid conditions are returned to their original values prior to the start of the activity where the exception occurs.

Multi-agent conversation will be examined in exception handling process either automatically or by user intervention. A snapshot of the exception handling peer user interface is shown in Fig. 4. Initial prototyping of the system has been carried out using Java coded software agents to provide an efficient workflow engine for the run-time function of the system. The P2P network environment of this prototype is based on Sun Microsystems JXTA [23]. XPDL (XML Process Definition Language) is used for process definition as it offers portability between different process design tools [21].

6 Conclusions and Future Work

This paper has proposed a Multi-agent based P2P Workflow Management System. P2P systems and multi-agents cannot replace each other for workflow application, however, they plug into each other to take advantage of both. The P2P network will be the frame where agents can work as distributed workflow engines, monitor workflow instance and react to unforeseen events. The feasibility of this work is being examined by implementing proof-of-concept prototype. Further work is required to validate and verify the ideas, the architectures and the techniques presented in this paper. This will include the completion of the underlying Multi-agent based P2P workflow architecture and their applications to a number of workflow problems and a verification and validation of the system, consisting of both research and experimental analysis.

References

1. Aldeeb, A., Crockett, K. and Stanton, M.: An Adaptive P2P Workflow Management System. In Proceedings of the 9th International Conference on Enterprise Information Systems (ICEIS-2007), pp. 428-433, Funchal, Portugal (2007)
2. Aldeeb, A., Crockett, K. and Stanton, M.: An Inter-organizational Peer-to-Peer Workflow Management System. In Proceeding of the 10th International Conference on Enterprise Information Systems (ICEIS-2008), pp. 85-92, Barcelona, Spain (2008)
3. Montagut, F., Molva, R. and Golega, S.: The Pervasive Workflow: A Decentralized Workflow System Supporting Long-Running Transactions. IEEE Transactions on Systems, Man, and Cybernetics – Part C, Vol. 38, NO. 3 pp. 319-333 (2008)
4. Yan, J., Yang, Y. and Raikundalia, G.: SwinDew- A P2P-Based Decentralized Workflow Management System. IEEE Transactions on Systems, Man, and Cybernetics, 36, No. 5 pp. 922-935 (2006)
5. Mo, G and Zhao, W.: Multi-Agent Interaction Based Collaborative P2P System for Fighting Spam. In Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'06), pp. 428-431, (2006)
6. Buhler, P. and Vidal, J.: Towards Adaptive Workflow Enactment Using Multiagent Systems. Information Technology and Management, 6, pp. 61-87 (2005)
7. Zhang, H. and Lesser, V.: Multi-Agent Based Peer-to-Peer Information Retrieval Systems with Concurrent Search Sessions. . International Conference on Autonomous Agents and Multi Agent Systems. Hakodate, Japan. ACM 1-59593-303-4/06/0005, pp. 305-312 (2006)
8. Fakas, G. and Karakostas, B.: A peer to peer (P2P) Architecture for Dynamic Workflow Management. Information and Software Technology, 46, pp. 423-431 (2004)
9. Hagen, C. and Alonso, G.: Exception Handling in Workflow Management Systems, IEEE Transaction On Software Engineering, 26, no 10, pp. 943-958 (2000)
10. Singh, M., Huhns, M.: Multiagent Systems for Workflow. International Journal of Intelligent Systems in Accounting, Finance & Management 8, pp. 105-117 (1999)
11. Miller, R. and Tripathi, A.: The Guardian Model and Primitives for Exception Handling in Distributed Systems. IEEE Transaction on Software Engineering, 30, pp.1008-1022 (2004)
12. Chen, Q., Dayal, U., Hsu, M. and Griss, M.: Dynamic-agent, Workflow and XML for E-Commerce Automation. Lecture Notes in Computer Science, pp. 314-323 (2000)
13. Sadiq, S. Orlowska, M. and Sadiq, W.: Specification and Validation of Process Constraints for Flexible Workflows. Information Systems, 30, pp. 349-378 (2005)
14. Zhang, H. Croft, W., Levine, B. and Lesser, V.: A Multi-agent Approach for Peer-to-Peer Information Retrieval Systems. International Conference on Autonomous Agents and Multi Agent Systems. New York, USA. ACM, 1-58113-864-4/04/0007, pp. 456-463 (2004)
15. Van der Aalst, W.: The Three Good Reasons for Using a Petri-net-based Workflow Management System. The Kluwer International Series in Engineering and Computer Science, Vol 428, pp. 161-182, Kluwer Academic Publishers, Boston, (1998)
16. Van der Aalst, W. and Basten, T.: Inheritance of Workflows: an Approach to Tackling Problems Related to Change. Theoretical Computer Science, 270, pp. 125-203 (2002)
17. Cai, T., Gloor, P. and Nog, S.: A Workflow Management System on the Web Using Transportable Agents. Technical Report PCS-TR96-283 (1996)
18. Müller, R., Greiner, U. and Rahm, E.: AgentWork: A Workflow System Supporting Rule-based Workflow Adaptation. Data & Knowledge Engineering 51, pp. 223-256 (2004)
19. Wang, M, Wang, H. and Xu, D.: The Design of Intelligent Workflow Monitoring with Agent Technology. Knowledge-Based Systems, 18, pp. 257-266 (2005)
20. Cao, J., Yang, J., Chan, W. and Xu, C.: Exception Handling in Distributed Workflow Systems Using Mobile Agents. In Proceedings of the 2005 IEEE International Conference on e-Business Engineering (ICEBE'05) (2005)
21. XPDL Support & Resources, http://www.wfmc.org/standards/xpdl.htm
22. Workflow Reference Model, http://www.wfmc.org/standards/referencemodel.htm
23. Sun Microsystems JXTA, https://jxta.dev.java.net/

Exploring the boundaries of Enterprise Information Systems

Malihe Tabatabaie, Richard Paige and Christopher Kimble

Computer Science Department
University Of York
Heslington, York, UK. YO10 5DD
{Malihe.Tabatabaie, Richard.Paige, Christopher.Kimble}@cs.york.ac.uk

Abstract. Around the globe companies are steadily becoming more connected. Therefore, there is an increasing demand for IT systems that can support large interconnected organisations. One of the main requirements of such systems is the ability to support business processes as well as organisational connections. There is a tension between how IT systems that support large interconnected organisations are currently built and organisations' functionalities. In addition, there is a need for such systems to have the flexibility to address new business requirements and opportunities. These tensions are addressed in the field of Enterprise Information Systems (EIS). The focus of this paper is on exploring the boundaries of the concept of Enterprise Information Systems through a survey of the literature on EIS. The aim is to delineate what might legitimately be described as an EIS and where this technology might be applied.

Keywords: Enterprise Information Systems, Organisation, Business Model, Business Processes, Goals.

1 Introduction

The development of software systems, that will meet the needs of various types of stakeholders, is the aim of most software engineering projects. However, with improving network technology and the growth of national and international markets, small and medium size companies link together to create large interconnected organisations. For example, in the case of Mitsubishi [1] 202 companies that are working in 30 different industries around the globe such as Banking, and Insurance, Mining are officially introduced in Mitsubishi's portal. Moreover, this portal explains that around 400 companies around the globe contain Mitsubishi name and more are working under the arch of Mitsubishi even though they do not have the Mitsubishi name.

These organisations need to connect various business processes' parts together. Also they need to be flexible enough to satisfy the new requirements of each part individually and the requirements of an organisation as a whole. Therefore, systems that can satisfy different stakeholders in large interconnected organisations are of increasing demand.

Enterprise Information Systems (EIS) is a type of system that has evolved as a result of this market demand. The aim of EIS is to support the requirements of large interconnected organisations. Nevertheless, this type of system still suffers from unclear definition. In addition, the boundaries of what constitutes an EIS must be clarifies, in order to make this definition more precise, and before starting any design and development. This paper discusses available definitions for EIS as well as the various types of EIS, based on the types of organisations that make use of them. Categorizing different types of organisation based on their *goals* will help us define the role of goals in design and development of EIS. The aim of this paper is thus to debate the domain and objectives of EIS; the result of these discussions leads to a proposed definition for EIS.

2 State of the Art Definition

To understand EIS requires us to first understand the notion of *enterprise*. The history of enterprises starts after the First World War, when previously independent organisations started to merge and coordinate their activities to increase the variety and amount of their products, and to introduce new industries to satisfy market demands [2].

Enterprise[1] is "an organisation created for business ventures, hence, growing enterprise must have a bold leader" [3]. This definition is an example of expressions that implies the close relationship between the term of enterprise and business. In various sources such as the definition given in Jessup [4], the definition of enterprise is strongly linked to business or business model concepts. Figure 1 illustrates the general structure of a business model. The business model includes business processes such as BP2 and BP3 in Figure 1; and business functions such as BF3 and BF4 in Figure 1. Business processes are "a set of logically related tasks performed to achieve a defined business outcome" [5]. The business processes for example, in the case of BMW, is putting new orders for part suppliers. When there was a new demand for specific model of BMW car such as Z5, this new market request is a business event that triggered a set of business, processes such as increasing the amount of resources for producing the Z5, and putting new orders in place to parts. Each of these business processes was subdivided into different business functions. Examples include the functions required for inputting new orders such as checking the part suppliers' ability for new demand, organizing the time that is needed for each part to arrive to assembly line, etc.

Some examples of current and modern enterprises are Mitsubishi, General Electric, and Boeing. according to [6] "business processes must be modelled and aligned across function, data and information systems that implement the processes". Therefore, the business function in this research refers to the functionality that is required for implementing a business process. Figure 1 is a simple explanation for a business process model. The aim of this diagram is mainly to explain business process and functions in a general business model. Each of these business functions can

[1] In some cases, the terms *enterprise* and *conglomerate* are used interchangeably.

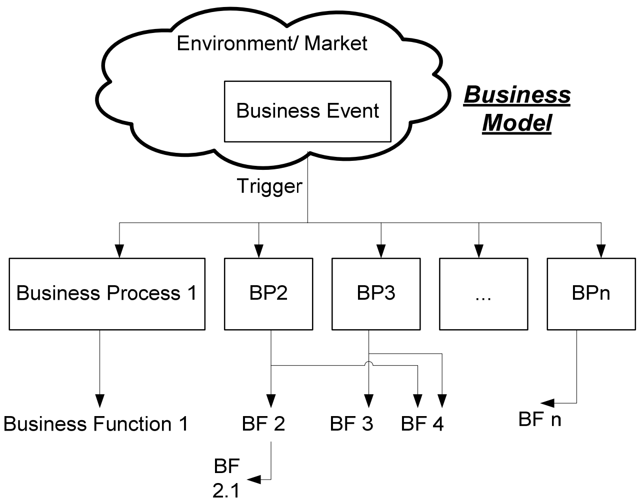

Fig. 1. Business Model (based on [6]).

trigger a business process. Moreover, the business processes can trigger other business processes. To keep the diagram simple, these cases are not shown in Figure 1. Understanding business models is helpful for developing EIS because the role of EIS is to integrate different business processes [7]. However, it seems that Legacy systems were the type of system that was used for the same propose. The reason for this claim is that before the concept of EIS have been defined, Legacy systems were the type of systems that were developed to handle the requirements of organisations [8]. A legacy system for this research is an existing computer system or application program, which continues to be used because the company does not want to replace or redesign it [8]. However, this type of system became expensive to maintain, therefore some of researchers in this area such as Strong [7] introduced EIS as a replacement for legacy systems. EIS creates another type of system that can be flexible enough to handle the new requirements of organisations based on the changes of a market. In contrast to enterprise systems, legacy systems are not designed to communicate with other applications beyond departmental boundaries [4].

In short, the common idea in existing definitions of EIS illustrates that an EIS amalgamates concerns from the various businesses, business processes, organisations, information systems, and information that circulate across an enterprise. In other words, it is about the business models of the organisation. However, a definition for EIS that just emphasizes the financial profit side of businesses for organisations is out of date. The reason for this claim is that EIS can be used in other type of organisations such as public and not for the profit organisations which their aim is not achieving financial profits but it is to provide better services. Next section will explain this type of organisations as well as private organisations in more detail.

3 Different Type of Enterprise Information Systems

The EIS definitions extracted from the literature link EIS to organisations [7, 9] or large companies [4] that we assume they refer to organisation. Based on this evidence, it can be said that different types of organisation can influence different type of EIS. Therefore, this section discusses categorizing organisations based on their goals. There are different ways of grouping organisations. in particular, Buck [10] categorises organisations in three groups:

- Public Organisations
- Private Organisations
- Not for Profit Organisations

The public organisations include central or local government, where elected members (e.g., minister) decide on the goals of the organisations, and may influence how goals will be achieved. The aim of this type of organisation is to supply services to or for the public, considering a ‘value for money’ rule. Examples of this type of organisation can be health service, prison, police, social security, environmental protection, and the armed force.

Private sector organisations are owned by individuals or other private organisations. This group of organisations can have the following goals:

- Satisfy their customers
- Satisfy their staff
- Satisfy their owners

All the above goals related to Private sector consider increasing the market demands for products or services.

‘Not for the profit’ organisations can be charities, mutual societies, etc, which provide some services for the society. The customers are also the member of the mutual society; therefore, they are the owner of the business. The value for money rule does exist in this group too. The way to evaluate the success of this group of organisations is to measure how well their goals are achieved considering the available resources.

Based on the information that was discussed in this section we create Table 1 as an example that illustrates some of the characteristics of organisations that could be described, it also summarizes the different type of organisations.

Table 1. Organisations' Categories.

Type of Organisation	***Decision Makers***	***Value for Money***	***Owner***	***Goal(s)***	***Example***
Public	Elected members	Yes	Public	Supply Services to or for the Public	UK central Government
Private	Share holders	No	Share holders	Satisfy customers/ Satisfy staff/ Satisfy owners	Mitsubishi
Not for Profit	Elected Manager	Yes	Members/ Customers	provide some services for the society or members	NCH (Children Charity)

By understanding the categories of organisation, we can focus on understanding their goals. By knowing the goals of organisations we can design and develop an EIS that satisfies the defined requirements and goals; but there is another question in this area: what are the EIS goals? Are the goals of EIS similar to the goals of organisations? It seems that EIS goals could be a sub domain of organisations' goals. When EIS goals get closer to the goals of organisations it could be a better EIS. Whilst EIS can improve the goals of organisation and could lead to the current goals to achieve better future goals, then EIS achieved their optimistic goals.

4 Enterprise Information System Goals

The domain and objectives of EIS is a debatable area. Therefore, categorizing EIS goals is controversial too; the reason for this claim is mainly the unclear definition for EIS. Even though, there are interests in defining this term as a practical concept for the organisations in academy and industry, however, in the real world, any large, middle, or even small companies introduce themselves as an enterprise. There is no clear line between a company and enterprise. Some terms such as company, organisation, conglomerate, etc have their own legal definition but in the case of enterprise, the legal definition does not exist. Real world organisations use a variety of approaches for developing different type of information systems, which can be EIS. Even though they might have developed EIS, it does not seem that they have a clear definition or understanding for enterprise; for example in the interviews that I had with people from the industry or researchers from academy, none had a clear definition for enterprises. In most cases, the enterprise definition refers to a large company. However, how large and what large means in this case is a challenge by itself.

The result of reviewing the literature and interviewing some of the stakeholders for understanding the definition of EIS (more information is provided in my qualifying dissertation), brings the idea of defining the goals of system. Defining the goals can be one of the first steps for developing a better system. By defining the goals of EIS, developing approach for systems can be clearer. As it was mentioned in section 4, organisations were categorized based on their goals. Nevertheless, this categorization was very high level and general but it shows the importance of defining the goals for developing better systems. Based on different views and goals, organisations can have different categories. At this level, the aim of the categorization was to show the importance of understanding the goals of organisations and as the result understanding the goals of EIS for developing them. There are various techniques to extract the goals of systems and describe them for stakeholders, techniques such as GSN [11]. The result of review on the literature illustrates that goal oriented techniques are mainly used in requirement engineering area and in some cases for software assessing. Moreover, we could not find a paper that discusses the goals of enterprises and suitable techniques for defining and documenting their goals. Therefore, it shows the open gap in this area of research.

5 Enterprise Information System Definition

In section 4, we discussed one of the challenges of working with EIS, which is an unclear definition. Moreover, in section 2, we discussed the current definitions for EIS, as the result of this review; the role of organisations emphasized in section 3. This section will present the result of this analysis about the definition of EIS. Our findings show that the following definition could address the challenges that were discussed in earlier sections.

EIS is a software system that integrates the business processes of organisation (s) to improve their functioning.

Integration of business processes plays an important role in this definition. Integration could be accomplished by providing standards for data and business processes. These standards will be applied for various part of the system such as database or clusters of databases. As one result of integration, information may then flow seamlessly.

Another point in this definition is the software characteristics of EIS. At this stage, we consider EIS as a type of IS; therefore this software system includes both humans and hardware.

The next term, used in the definition is *organisation*. Different types of organisations were discussed earlier in this paper. Organisations may include an organisation with its partners, or a group of organisations. Table 2 refines the above definition and describes what we propose as the objectives, goals, domain, and challenges of EIS.

Table 2. EIS boundaries, objectives, and challenges.

Objective	Integrity of the organisation and collaborators
	Seamless Information flow
	Suitable access to data and information for various stakeholders
	Matching the software system structure with organisation structure
Goal	Improving coordination, efficiency, and decision-making of business process in an organisation
Domain	Covers the internal and external business activities of organisation
Challenge	Security challenges that should be considered carefully for organisations' processes. Otherwise, mixing the required information of one business process with another one can cause problem for the organisation
	Improve flexibility in organisation processes

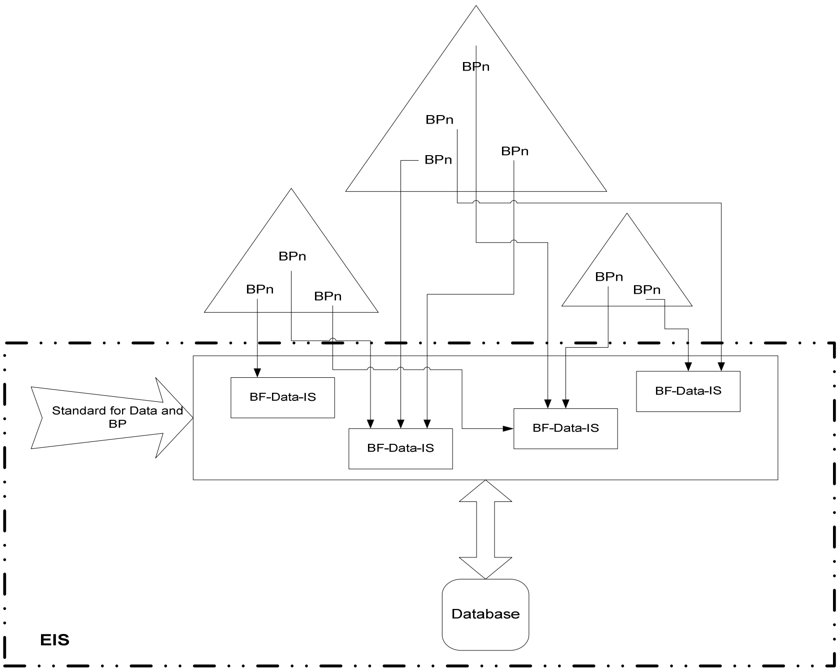

Fig. 2. Enterprise Information System

In addition, Figure 2 describes the definition of EIS graphically. Note that BP in this Figure means business processes; each organisation contains various business processes. In Figure 2 the database could be a cluster of databases, but in the end the system should have a interface with data without having a concern about where they are and what are the various resources. As can be seen in this Figure, the dashed line is the gray boundaries of EIS.

6 Conclusion

In conclusion, the aim of this paper was to illustrate the objectives and domain of EIS by providing a discussion on influential elements such as goals and organisations. The result of our research in this stage proposed a definition for EIS that was discussed in detail in section 5. There are various definition mainly based on business with financial profits; but as it was discussed in section 3, EIS belong to various type of organisations. Therefore, definitions that only consider businesses with financial profits are out of date. Defining EIS and discussing the domain and objectives of EIS is the essential steps for introducing this term to academic and industrial researchers. Providing suitable definition for EIS open the gates for future researches on how to improve development of this type of systems. The final aim is to develop better system for organisations that can satisfy their changeable requirements [11].

References

1. Mitsubishi: About Mitsubishi. http://www.mitsubishi.com/e/group/about.html. (2007)
2. Fruin, M.W.: The Japanese enterprise system. New York: Oxford University Press. p. 397 (1992)
3. WordNet, A lexical database for the English language. http://wordnet.princeton.edu/perl/webwn?s=enterprise (2007)
4. Jessup, L. and Valacich, J.: Information Systems Today, Why Is Matters. Second Edition, New Jersey: Pearson Education Inc. (2006)
5. Davenport, T.H.: The new industrial engineering: Information technology and business process redesign. Sloan management review (1990)
6. Kaisler, S., Armoiur F. and Valivullah M.: Enterprise Architecting: Critical problems. Hawaii International Conference/ IEEE (2005)
7. Strong, D.M. and Volkoff O.: A roadmap for Enterprise system implementation. IEEE Computer Society, pp. 22-29 (2004)
8. Robertson, P.: Integrating legacy systems with modern corporate applications. Commun. ACM, **40**(5): pp. 39-46 (1997)
9. Laudon, J.P. and Laudon K.C.: Management Information Systems: Managing the Digital Firm. 10th Edition, Prentice Hall (2007).
10. Buck, E.: Different Types of organisation. NEBS Management/QMD Ltd. (2000)
11. Kelly, T., et al.: The goal structuring notation: a means for capturing requirements, rationale and evidence, in Requirements Engineering at the University of York; Yellow Report, Vickers A.J. and Brooks L.S., Editors. University of York (1998)

Part III

Architectures and Models

Evaluating Multiprocessor Architectures for Image Processing Applications

M.J.Freeman

Department of Computer Science, University of York
mjf@cs.york.ac.uk

Abstract. Embedded image processing applications require a complex, conflicting mix of requirements, matching processing performance to tight power and cost constraints. In order to meet these real time processing deadlines, application specific system architectures must be used. This paper examines the use of multiprocessor based design flows for this application domain, through the development of an embedded background subtraction hardware core used within a car classification system.

1 Introduction

System-on-Chip (SOC) designs, have become a standard design approach in many digital systems, integrating system components: processor cores, memories and custom logic blocks within a single device. This has become possible with the development of process technologies of 0.18µm and below, allowing 10s of millions of transistors on a chip. To fully exploit the potential of these transistor rich devices there has been a move to hardware description languages (HDLs), allowing hardware components to be automatically synthesized from textual descriptions. The introduction of HDL logic and behavioural synthesis in the late 1990's has increased designer productivity allowing more of a design to be inferred from the HDL instead of being explicitly stated. Unfortunately, even high level HDL synthesis has not proved sufficient for modern SOC designs, resulting in a productivity gap between the number of transistors available and the complexity of the systems that can be designed and implemented.

To deal with this ever increasing complexity hierarchy is exploited, moving to higher levels of abstraction, reducing complexity in terms of the number of objects handled at any one time. This can be seen in modern design flows with a move away from board level designs: standard components and application specific integrated circuits (ASIC) to SOC, multiprocessor SOC (MPSOC) and network on a chip (NOC) designs. Traditional SOC design solutions were commonly based around a single processor with dedicated data processing cores. With increasing transistor resources available to the designer these data processing cores are being replaced with standard processor cores to reduce development time and increase flexibility i.e. MPSOC designs are quicker to implement, debug and test due to their higher software content than dedicated hardware. However, replacing application specific hardware components with general purpose processor based solutions can reduce processing performance, therefore task and data level parallelism within the processor module need to be exploited in order to meet real time processing requirements. This approach results in a hardware / software co-design problem, partitioning a system's specification into hardware and software components. Balancing inexpensive and flexible software solutions with high speed hardware, such that only that hardware that is required to meet timing is implemented in dedicated hardware.

The case study chosen for this work is a background subtraction algorithm based on an adaptive Gaussian mixture model [Zivkovic Z., 2004], used as part of a car classification system. The development platform used to prototype this design is a Xilinx Spartan 3 [Xilinx, 2008a] field programmable gate array (FPGA). The remainder of this paper will focus on how MPSOC design solutions can be applied to embedded image processing applications, in particular a reconsideration of the type and role of the processor used in modern FPGA designs.

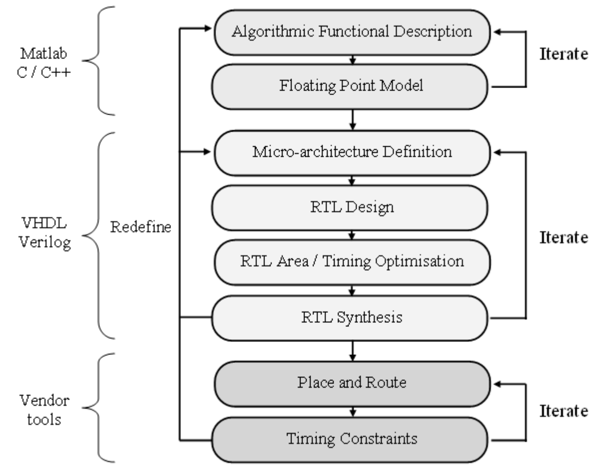

Fig. 1. Typical FPGA design flow

2 FPGA System design

With the continued development of direct multi-media support within desktop PCs a number of image processing algorithms, once the preserve of dedicated hardware or processor arrays, can now be achieved relatively easily in off the shelf systems. The ease and availability of desktop processing power contrasts dramatically with embedded solutions. Designing systems for embedded image processing applications offers a conflicting set of requirements. Such systems generally require high levels of processing performance to achieve specified frame rates, working on large image data sets producing intermediate data objects of comparable sizes. All of this must be achieved in a minimal hardware solution in order to meet power and cost constraints associated with embedded battery powered devices. A typical FPGA design flow for such systems is illustrated in figure 1. Initial prototyping and algorithm selection is normally performed in a rapid prototyping environment such as Matlab [MathWorks, 2008]. The relative merits of different techniques can be quickly compared and a C++

implementation produced to assess its performance i.e. accuracy and memory requirements. This prototype is then mapped onto an appropriate system architecture before being coded in a HDL. At this stage HDL simulations can be performed to estimate processing performance. If the specified processing performance is not achieved the system architecture must be modified to improve performance. When the specification is fulfilled the HDL is synthesized into a netlist that can be downloaded into the FPGA, configuring its internal elements to perform the desired function.

Modern system level design techniques make extensive use of pre-designed, complex components that can be attached via a standard interface to a system bus e.g. processor cores, minimizing the amount of new hardware that needs to be designed and tested [Microsoft, 2007]. These components are often referred to as Intellectual Property (IP), typically bought in from a number of different external suppliers. While the idea of IP reuse promises great benefits, matching an IP core to the functional and performance constraints defined in the specification is not always possible. This results in a compromise, selecting one component from a set of matching components which best meets the design goals with the minimum number of modifications. MPSOC design flows have therefore resulted in a trade off, simplifying the design process by replacing complex hardware accelerators with equivalent software implementations at the cost of significant increases in on-chip memory i.e. local processor instruction and data memories, shared memory and queues etc. As a result on-chip memory consumes more silicon area than any other type of circuit within a modern IC. This requirement effectively reduces the maximum processing performance of the IC i.e. memory is an enabling component, reducing the silicon area available for active processing elements such as adders, multipliers etc. The main focus and hypothesis of this work is that a key requirement in designing MPSOC based systems is that the amount of on-chip memory must be minimised through a revaluation of the processor's role, using hardware / software co-design techniques to move common, static software components into dedicated hardware.

3 FPGA based processor selection

To maximize performance MPSOC designs must identify and exploit parallelism within the SOC architecture through both computation and communication means. Increasing the number of processors on a chip places particular importance on the communication structures used. Standard system buses don't scale well in a FPGA. One factor is the increased routing complexity, however the main factor is the increased logic depth caused by the associated address, data and control bus multiplexers i.e. the more components attached to a bus the lower its performance will be. In general communications models fall into two categories, explicit and implicit. Explicit are handled by send and receive commands e.g. message passing through channels etc and implicit through shared memory. Message passing is the preferred communication structure for systems constructed from a set of loosely coupled, largely independent tasks. For more tightly coupled tasks a shared memory data structure is normally preferred i.e. a block of memory on a common data bus implementing both handshaking flags and data storage. Another important consideration in MPSOC design is the programming model used. The most efficient in terms of hardware requirements is a single threaded task, significantly reducing the required memory footprint. However, not all algorithms fit this programming model, therefore to simplify software development a real time operating system (RTOS) is typically required. A RTOS is a complex collection of software components: kernel, management of system resources, memory protection, facilitating communications among software tasks, multitasking scheduler, interrupt exception signal handling etc. However, when compared to the memory and processing requirements of the application code these RTOS software components can significantly increase a processor's memory and processing requirements i.e. the more layers of software that are required to process an item of data, the higher latency and lower bandwidth will be. Therefore, to support the application code and the RTOS library more on-chip memory will be required and a more complex processor architecture must be used in order to meet any real time deadlines [Kohout et al, 2003].

In general embedded applications do not take full advantage of a general purpose processor's (GPP) broad capabilities, resulting in wasted silicon that could have been more productively used. These issues have lead to the development of configurable or extendable processors allowing application specific instructions, memory and communication architectures to be defined. These modifications however do come at an increased hardware cost, increasing the complexity of the instruction decoder, functional units and the internal bus architecture within the processor e.g. to allow multiple operands to be fetched on the same cycle etc. These issues can have a significant impact on the processor's maximum clock speed reducing the performance of other sections of the program. Therefore, the approach taken in this work is to move the application specific data processing functionality out of the processor into a dedicate co-processor, accessed through a remote procedure call (RPC) interface. System functions are now implemented as a set of software objects managing the flow of data between the processor and its co-processors. This design emphasis goes beyond traditional approaches, reducing the processor down to its core functionality: control structures (selection, branches etc), event and task management, address and pointer management, which are applicable to all applications. This allows a simpler processor architecture to be used, maximizing its clock speed and making its real time performance predictable i.e. removing pipeline and cache complexities. This shift in data processing enables the processor's word size to be reduced e.g. 32bit to 16bit, 8bit or application specific sizes, significantly reducing on-chip memory and hardware requirements. Reductions in instruction and data memory widths can now be used to increase processing performance through replication of these processing nodes i.e. a single 32 bit processor will have an equivalent memory footprint to two 16 bit or four 8 bit processing cores. Taking this approach increases the number of processors available to the designer increasing the possible parallelism within the MPSOC design.

Processor selection is highly dependent on data processing requirements, communications and RTOS overheads. To reduce on-chip memory requirements the standard processor module shown in figure 2 has been developed. The RTOS has been implemented directly in hardware, minimizing instruction code size and associated processing overheads and increasing parallelism within the system, enabling a very simple processor architecture to be used, in the case a Xilinx

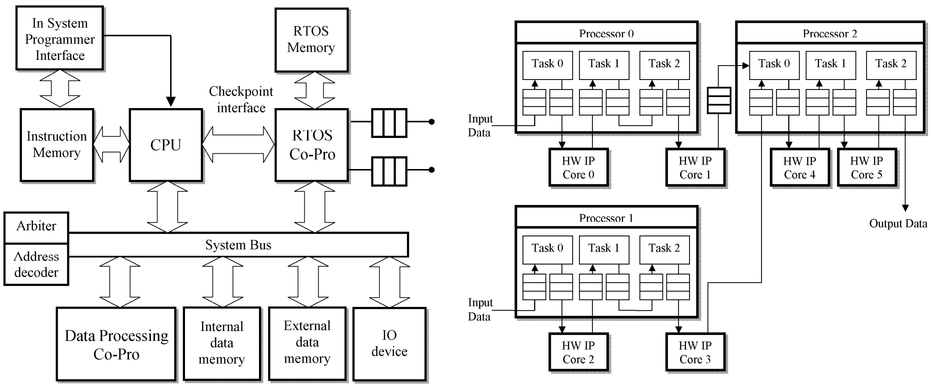

Fig. 2. Standard processor module (left), MPSOC architecture (right)

PicoBlaze [Xilinx, 2008b]. The RTOS co-processor directly supports delays, signals, interrupts, semaphores, shared memory, pipes (software and hardware) and task management (fixed priority scheduling). The aim of this architecture is to move common, static functionality out of software into hardware, therefore reducing instruction related on-chip memory and off-chip memory accesses. The MPSOC functionality is implemented from a number of these modules communicating through shared memory or communication channels i.e. either as parallel units or elements within a hierarchical dataflow architecture, as shown in figure 2.

4 Case study : Background subtraction

The case study used to evaluate this work is a background subtraction algorithm based on an adaptive Gaussian mixture model. Each pixel's RGB colour components are modeled by a series of Gaussians, these being continually updated to compensate for lighting changes, slow moving objects and long term changes to the background scene. When a new object of interest is introduced into the scene these RGB values fall outside the previously modeled distributions allowing the algorithm to classify these pixels as being associated with a foreground object, as illustrated in figure 3. The main disadvantage of this algorithm is its computational complexity, requiring a significant number of multiply, divider and accumulate operations. Processing can be improved by down sampling the image, then re-scaling the mask back up to the original image size for classification. This technique does significantly reduce the amount of data that needs to be processed, however, this is at the cost of accuracy, complicating foreground classification, as shown in figure 4. It is therefore desirable that improvements in processing performance come from task level parallelism through processor replication and data level parallelism within the data co-processor.

The background subtraction algorithm is implemented on a per pixel basis making it easily parallelizable at the task level. The image is divided into a number of equally sized segments, each segment being assigned to a separate processor. The number of processors being determined by the available FPGA resources. This processor array is implemented using the processing module shown in figure 5. Each module contains two processor cores, allowing them to share the same instruction memory i.e. halving the instruction memory footprint. Image data and working data are stored in an external DDR2 SDRAM memory module. From HDL simulations localizing bus traffic is a key design step in maximizing performance i.e. accessing external memory requires access to one or more system buses, increasing latency when compared to accessing data from its local bus and memory. To maximise external memory availability and bandwidth only block transfers are therefore made to and from local scratchpad memory, reducing the significance of control overheads associated with SDRAM accesses. These transfers are implemented in the system bus bridges using direct memory access (DMA) controllers with both read and write buffers to decouple the processor from these transfers.

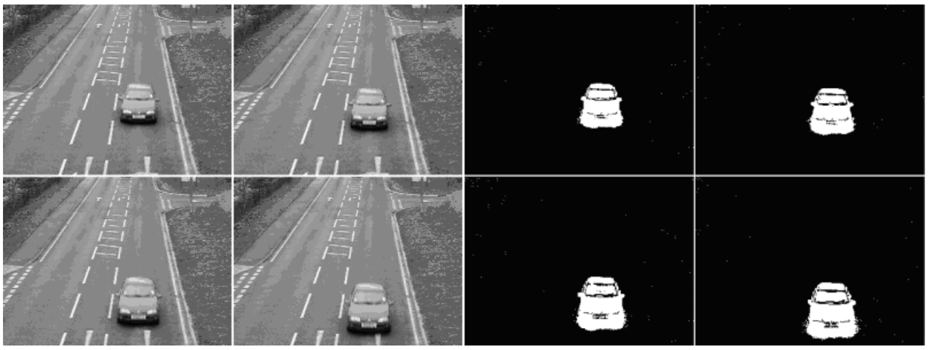

Fig. 3. Background subtraction, camera image (left), generated mask (right)

Fig. 4. Down sampling effects, 320x240 (right), 160x120(middle), 80x60 (left)

Moving application data processing out of the processor allows its architecture to be optimized to minimize on-chip memory. However, this potentially just moves this requirement into a new application specific hardware component. To prevent a new co-processor having to be designed for each application a configurable co-processor architecture has been designed that can be easily integrated into the envisaged MPSOC architecture, as shown in figure 6. Data is passed to the co-processor through two input channels having: command (specifying desired operation, feedback path, local memory storage address) and operands (data to be processed). Channel A is passed a command and two operands, channel B is passed a command and one operand, the second being generated by internal functional units. When the issue controller detects that a command can be executed it is assigned to a functional unit i.e. adder, multiplier, divider or square root unit. The number of functional units can be configured through HDL parameters to match the available FPGA resources. On completion a result can be passed to the output channel, feedback as a channel B input or accumulated. From HDL simulations overlapping memory accesses and functional unit execution is another key design step in maximizing performance e.g. as model data is written to local memory it is also passed to the data co-processor generating intermediate results, buffered in local memory and used by the processor on the

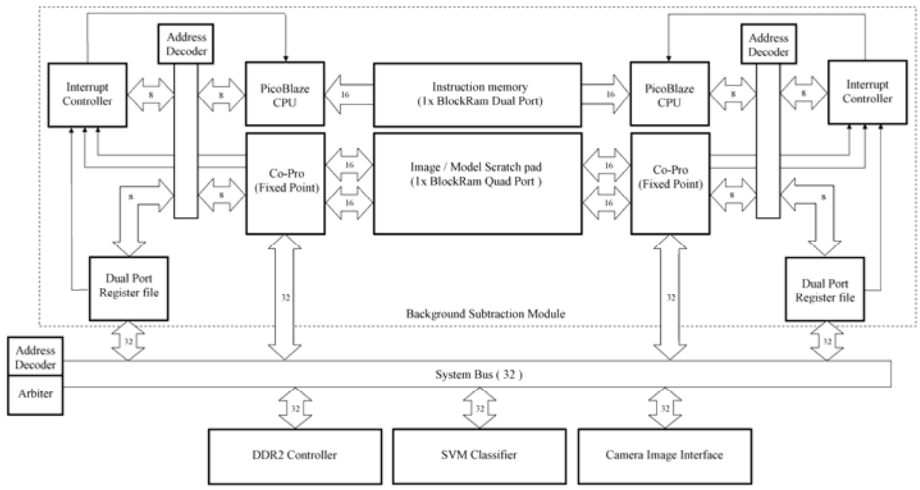

Fig. 5. Dual core pixel processing module

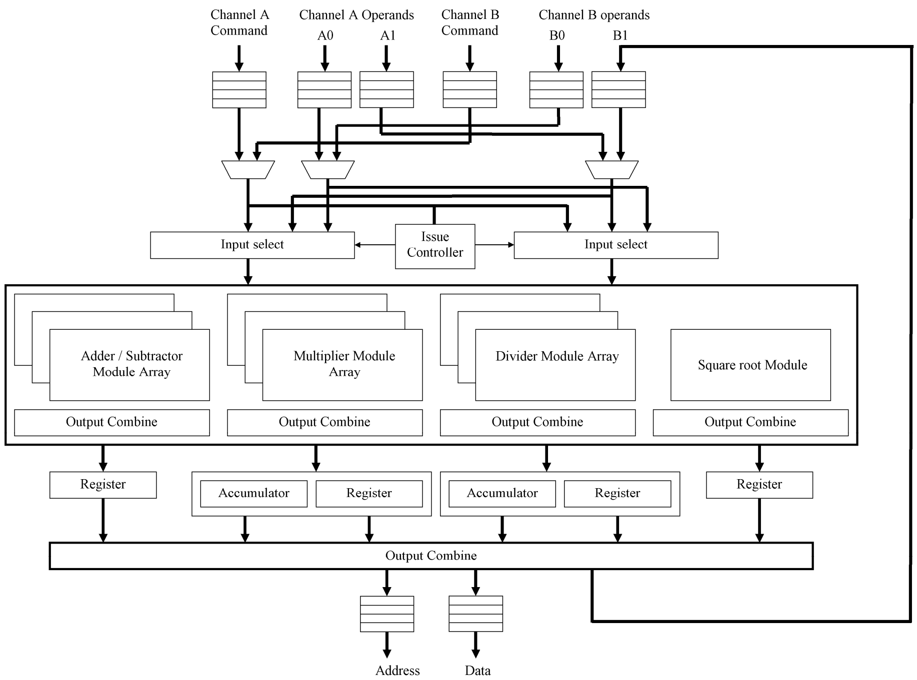

Fig. 6. Data processing co-processor

completion of this data transfer. As external memory is a shared resource, structural hazards will occur. Therefore, latency hiding and latency tolerance are important design considerations in the data co-processor design e.g. pre-fetching data from memory to hide memory latency, out of order instruction execution and deferred execution of those operations dependent on incoming data, switching tasks during long processor-processor or memory-processor communications delays.

5 Conclusions

Evaluation of different MPSOC background subtraction implementation is still ongoing. Initial results suggest that these design flows do offer viable FPGA design solutions through the use of replicated processor arrays. However, this is highly dependent on the size of the target FPGA and its available resources. The most significant of these being on-chip memory. Key issues to be resolved are the affects of external memory bandwidth and latency on the system's processing performance and how these limit the maximum number of processors that can be used.

Modern SOC designs will continue to move towards MPSOC based design flows in order to simplify design and testing complexities. However, the current trend of using powerful GPPs requiring significant hardware, on-chip memory and external memory bandwidth is unsustainable. To minimise external memory bandwidth and the affects of latency on processor performance a more considered design approach should be taken, revaluating the assignment of application and RTOS functions within the processor network. Therefore, it is proposed that MPSOC and IP re-use design flows should be combined, through the development of a new hardware architecture i.e. the standard processor module. This hardware supports common core software requirements for such systems, simplifying programming through the development of processor based parallel processing and dataflow networks. The main design flow is still software biased, however, this is now supported through the used of a set of standard hardware components. This shift allows selected data processing and RTOS functions to be moved out of software, increasing parallelism, minimising on-chip memory and therefore, reducing silicon area and power requirements.

References

1. Kohout, P., Ganesh, B. and Jacob, B., 2003. Hardware support for real-time operating systems. In Proc. of the 1st IEEE/ACM/IFIP Int. Conf. on Hardware/Software Codesign and System Synthesis, pp. 45–51, USA
2. MathWorks, 2008. Available at: http://www.mathworks.com
3. Microsoft, 2007. The Manycore shift. Available at: http://www.microsoft.com/presspass/events/supercomputing/docs/ManycoreWP.doc
4. Xilinx, 2008a. Spartan-3 FPGA Family Data Sheet. Available at: http://www.xilinx.com/support/documentation/data_sheets/ds099.pdf
5. Xilinx, 2008b. PicoBlaze 8-bit Embedded Microcontroller User Guide. Available at: http://www.xilinx.com/support/documentation/ip_documentation/ug129.pdf
6. Zivkovic, Z., 2004. Improved Adaptive Gaussian Mixture Model for Background Subtraction. In Proc. of the 17th Int. Conf. Pattern Recognition, pp. 28-31, UK

A Hierarchical Metapopulation Model for Disease Dynamics Built on Population Movements of Both Patch-Coupling and Migration

Xiaosi Wang and Calvin Dytham

Department of Health Sciences and Department of Biology, University of York
xw502@york.ac.uk
cd9@york.ac.uk

Abstract. In the past, frequent movements and migration were studied separately in metapopulation models. The levels that exist in a hierarchical metapopulation model were also limited to two and only recently increased to multiple levels. Moreover, a generalisable deterministic model was not available. Here we introduce a novel model incorporating both movement scenarios as well as a multiple-level structure. We describe the system in simple differential equation form. The simulations of different distributions of local contact rates for disease transmission suggest that local information is important for predicting disease dynamics. The comparison between the results from a solely migration-based multilevel model and the model discussed in this paper suggests that diseases with low transmission rates can spread rapidly and infect a large number of susceptible individuals in a short time if they appear in a population where frequent movements are dominant.
Keywords: multilevel system, patch-based system, disease modelling, epidemic, mathematical epidemiology

1 Introduction

It is now clear that understanding and prediction of the progress of disease requires consideration of the spatial arrangement of individuals and many recent computational models of disease spreading place a strong emphasis on the role of the spatial heterogeneity of human populations [1–3]. It is clear that the distribution of individuals within a population significantly affects the process of pathogen dispersal, but the exact operating functions linking transmission and spatial pattern are still unknown. However, even when using pathogen-related parameters that are invariant, e.g. a model where pathogen infectivity and virulence are kept constant throughout the simulations, the results generated by spatial models simulate the real epidemics very well [2, 4–6].

There are various ways to integrate spatial heterogeneity into models. One popular method is to build metapopulation models, where the population is divided into a network of smaller subpopulations on patches [2, 7, 8]. Individuals within each subpopulation are assumed to be well mixed. Metapopulation models allow explicit mathematical expressions and straightforward numerical solutions [9], and hence play an important role in mathematical epidemiology. Hierarchical metapopulation models are a special type of general metapopulation models [2, 10]. They consider the hierarchy involved in human movements (i.e. that subpopulations have some non-random pattern of connections) [11, 12] and simulations from these models show that disease spreading is significantly influenced by multilevel movements [2, 6]. New studies based on real human mobility data also provide evidence to support the argument that individual movements occur at different levels [11, 12].

Another thing to consider is that the ways individuals interact with each other. Interactions between individuals involve both within-patch interactions and between-patch interactions. It is assumed that the contacts between individuals on each patch are frequent and hence random mixing applies within the subpopulation. Between-patch interactions are of more interest and usually modelled by two methods, depending on the frequency of movements. If the interaction between two patches is dominated by frequent movements (e.g. people commuting to and from work), the subpopulations are said to be interacting with each other in a way like particles randomly bumping into each other. In other words, any infection occurring on one patch has the force of infection on the susceptible individuals in the closely related patches [1, 13, 14]. The force of infection is defined as the per capita rate that the infected individuals transmit the disease to susceptible individuals [15]. Alternatively, if the movements between two patches mainly take the form of migration, it means that individuals migrate to the host population with the disease status they get from the home patch first and then take part in the disease transmission process in the host patch [1, 9]. These two scenarios were studied separately in the past [1,13]. In real populations, it is obvious that both scenarios occur simultaneously.

Here we build a metapopulation model based on multilevel movements including both patch-coupling and migration. At the lowest level, where the population movements between the patches are most frequent, the patches are coupled by the force of infection; while patches with less frequent movements in between are linked by migration. In this paper, these two kinds of patch relationships are referred to as close-related patches and not close-related patches. Moreover, it does not necessarily mean that well-connected patches are geographically close, in contrast to previous work [2]. Human mobility tends to be more complex than animal migration or plant dispersal and is not necessarily related to geographic distances [16, 17].

2 The Model

The hierarchical system we set up to describe the metapopulation model consists of L levels of movements. The number of patches at the same level is denoted by a fixed branching ratio B for simple indexing. Therefore the total number of patches is B^L, denoted by n. We assume that the fewer the movements between two patches, the larger level difference (D) between them. The level difference is the number of levels to reach the common ancestor node in the hierarchy. The level difference between close-related patches, i.e. the first level when counting levels, is defined as 0. One possible system is illustrated in Figure 1 as an example. Individuals are assumed to be homogeneously mixed within each patch. All patches are assumed to have identical within-patch population dynamics and environmental conditions [8]. Person-to-person contact, which leads to disease transmission, takes place when individuals meet in one of the n patches. Patches interact with each other through either coupling or migration. Initially the total population is distributed evenly across all n patches and one infectious individual is introduced to the system. Individuals in each patch are classified in terms of their infection status: susceptible, infected or recovered, within which the numbers of individuals are denoted by S, I and R respectively. The susceptible class includes all healthy people with no immunity to the disease, the infected class includes people who have caught the disease. For simplicity, infected individuals are assumed to be infectious immediately. R denotes the recovered group, with lifelong immunity [13, 15]. We model a non-fatal, communicable disease, such as the common cold or influenza virus, spreading much faster than the natural demographic process. Therefore our basic framework is the simple *SIR* model (Equation 1) [15, 18]:

$$\begin{aligned} \frac{dS}{dt} &= -\beta IS \\ \frac{dI}{dt} &= \beta IS - \gamma I \\ \frac{dR}{dt} &= \gamma I \end{aligned} \tag{1}$$

where disease is transmitted through person-to-person contact which is modelled by the adequate contact rate for disease transmission (β) and recovery rate (γ). This is the normalised form, i.e. all the variables are proportions. It models the pure disease transmission process within a homogeneously mixed population without considering other effects, such as demographic effects, spatiotemporal effects and so forth. In our system, where the interactions between patches take two different forms, we need to consider both effects. The force of infection for the simple *SIR* model is βI. To generalise it to the single patch in our model, we need to consider that the force of infection on patch i is affected by the sum of the infection situations on the patches close-related to it, i.e. $\sum\limits_j \beta_{j,i} I_j$, where $j \in J$, J: the set of indices of the patches close-related to patch i. In this model, we allow the adequate contact rate to vary between close-related patches [13] and therefore we can examine the effects of different contacting rules on disease dynamics. The range of values of β is consistent with that for influenza [15].

The other process that changes the number of infected individuals in patch is migration. Both emigration and immigration are assumed to operate between two patches which are not close-related. The per capita migration rates of infected, susceptible and recovered groups in the whole system are denoted by θ, ϕ and ξ respectively. The immigration rate of susceptible individuals from another patch to patch i is calculated by a function $\theta_{ki} = \theta Ce^{-CD}$, where $k \in K$, K: the set of indices of the patches linked to patch i by migration. The subscript ki indicates immigration and ik emigration. Ce^{-CD} is a normalised general exponential function. It is the simplest form for representing the mechanism that the migration rate decays as the level difference between two patches increases. C is a constant, which scales the function and D is the level difference (described above). Similarly, $\phi_{ki} = \phi Ce^{-CD}$ and $\xi_{ki} = \xi Ce^{-CD}$ describe the rates of movement for infected and recovered individuals respectively. Finally, all the immigrants from different patches are summed and all the emigrants to different patches are subtracted for each disease group to get the total proportion of immigrants and emigrants.

As we described above, because all the patches are homogeneous in population distributions, disease spreading behaviours and patterns of between-patch movements, we are able to express the population dynamics on an arbitrary patch *i* into a differential equation form (Equation 2):

$$\begin{aligned}\frac{dS_i}{dt} &= -(\sum_j \beta_{ji} I_i) S_i + \sum_k \theta_{ki} S_k - \sum_k \theta_{ik} S_i \\ \frac{dI_i}{dt} &= (\sum_j \beta_{ji} I_i) S_i - \gamma I_i + \sum_k \phi_{ki} I_k - \sum_k \phi_{ik} I_i \\ \frac{dR_i}{dt} &= \gamma I_i + \sum_k \xi_{ki} R_k - \sum_k \xi_{ik} R_i\end{aligned} \quad (2)$$

All migration parameters have very small values in this paper and therefore have little effect on the variability of local population dynamics during the simulation. All simulations are run until equilibrium is reached. Here we use two systems with the same number of patches in total: 1) branching ratio of four ($B = 4$) and level of three ($L = 3$); 2) branching ratio of two ($B = 2$) and level of six ($L = 6$).

3 Results

Figure 2 shows the time series of proportion of infected individuals for β being uniformly distributed and normally distributed. More fluctuations and different sizes and durations of the infection changes are observed with the uniform distribution. It shows smaller but longer epidemics in the uniform distribution based model than those when β is drawn from a truncated normal distribution.

In previous work using hierarchical metapopulation models, contacts between patches are based solely on migration behaviours [2,6] and comparison results with and without patch-coupling is shown in Figure 3. It clearly shows that patch-coupling accelerates the spread of the disease through the system and leads to more cases, even with a smaller contact rate for disease transmission.

4 Discussion

Combining close-related patches and migration-related patches, we obtain behaviours not observed in previous studies [2,6]. A uniform distribution of β means that we have little prior knowledge of the adequate contact rates on patches, so the random chosen value has equal opportunities to stay at any point within the lower and upper bounds. We have 64 uniformly distributed random numbers and we expect strong stochasticity. The results illustrated in Figure 2 confirm this. On the other hand, 64 normally distributed random numbers tend to surround the mean. Consequently we observe some randomness but still see the three-level pattern. In conclusion, estimating the adequate contact rates in real life is an important step towards choosing right models for predictions.

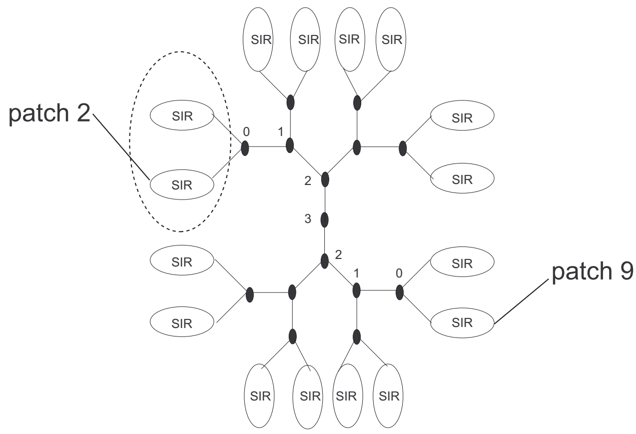

Fig. 1. One example of the multilevel system ($B = 2$, $L = 4$). The small ellipses represent the patches, the black nodes illustrate the different levels and the dashed ellipse represent one example of close-related patches. *SIR* means that the simple Susceptible-Infected-Recovered process applies within each patch. It also shows an example of calculating the value of D, the level difference between any two patches (see method for description). D = 3 for patch 2 and patch 9 (anticlockwise numbering) according to the levels they belong to (i.e. you have to move up three levels before these two patches share a common node).

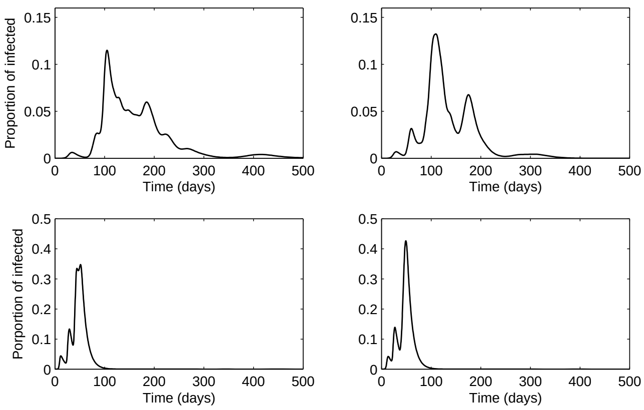

Fig. 2. The effects of two different distributions of β on disease spreading. $B = 4, L = 3$. (Top): two simulations based on a uniform distribution of β ($0.0 < \beta < 0.6$); (Bottom): two simulations based on a truncated normal distribution of β (mean = 0.3, standard deviation = 0.1, $0.0 < \beta < 0.6$). It shows that we get smaller but longer epidemic from the uniform distribution based model then those from the normal distribution. Moreover, the results from the uniform distribution are more stochastic.

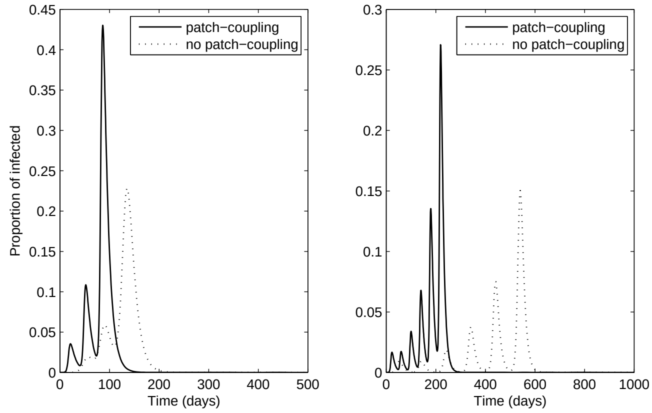

Fig. 3. Comparison between migration-based model and patch-coupling-migration-based model for two systems: $B = 4, L = 3$ (Left); $B = 2, L = 6$ (Right). For migration-based simulations (dashed line), $\beta = 0.3$ is used; whereas $0.1 < \beta_{ki} < 0.3$ is applied for patch-coupling-migration-based simulations (solid line), which means the adequate contact rate on average is smaller than three. It shows that even with a smaller contact rate, the size of epidemic is not reduced for patch-coupling-migration-based model. It also shows that the synchronisation of the system is more rapid in the patch-coupling-migration-based model.

It was shown by previous studies that patch-coupling is a quick way for the disease dynamics on each patch to synchronise [13, 14], but observing the effects in a multi-level metapopulation model was not realised. We obtained a larger epidemic with even smaller contact rates for transmission in a model including patch-coupling. Therefore it demonstrates that epidemics are likely to be much worse in a large population where movements are frequent between sub-populations. We suggest that health authorities cannot ignore an infectious disease with low transmission rate that occurs in a large population where people have more frequent short-trips between subpopulations.

We promote a model based on both frequent movements and long-lived travels and it shows that differentiated contact rates make the disease dynamics more complicated but still tractable. Since local information of contact rates are not usually collected [14], we expect that investigations on such data will be helpful both for validating the model and for facilitating better prediction of the spread of diseases through human populations.

5 Acknowledgements

We thank Graham R. Law and Kate E. Pickett for persistent support, Alun L. Lloyd, Roby Muhamad and Duncan J. Watts for answering questions. XW is funded by a studentship from the Department of Health Sciences at the University of York.

References

1. Lloyd, A.L., Jansen, V.A.A.: Spatiotemporal dynamics of epidemics: synchrony in metapopulation models. Math Biosci **188** (2004) 1–16
2. Watts, D.J., Muhamad, R., Medina, D.C., Dodds, P.S.: Multiscale, resurgent epidemics in a hierarchical metapopulation model. Proc Natl Acad Sci U S A **102**(32) (Aug 2005) 11157–11162
3. Volz, E., Meyers, L.A.: Susceptible-infected-recovered epidemics in dynamic contact networks. Proc Biol Sci **274**(1628) (Dec 2007) 2925–2933
4. Gandon, S., Mackinnon, M., Nee, S., Read, A.: Imperfect vaccination: some epidemiological and evolutionary consequences. Proc Biol Sci **270**(1520) (Jun 2003) 1129–1136
5. Boots, M., Mealor, M.: Local interactions select for lower pathogen infectivity. Science **315**(5816) (Mar 2007) 1284–1286
6. Wang, X., Dytham, C., Law, G.R.: The effects of multilevel metapopulation structure and migration kernels on disease spread. under review (August 2008)
7. Levins, R.: Some demographic and genetic consequences of enviromental heterogeneity for biological control. Bull. Entomolog. Soc. America **15** (1969) 237–240
8. Hess, G.: Disease in metapopulation models: Implications for conservation. Ecology **77**(5) (1996) 1617–1632
9. Jansen, V.A., Lloyd, A.L.: Local stability analysis of spatially homogeneous solutions of multi-patch systems. J Math Biol **41**(3) (Sep 2000) 232–252
10. Hufnagel, L., Brockmann, D., Geisel, T.: Forecast and control of epidemics in a globalized world. Proc Natl Acad Sci U S A **101**(42) (Oct 2004) 15124–15129
11. Brockmann, D., Hufnagel, L., Geisel, T.: The scaling laws of human travel. Nature **439**(7075) (Jan 2006) 462–465
12. González, M.C., Hidalgo, C.A., Barabási, A.: Understanding individual human mobility patterns. Nature **453**(7196) (Jun 2008) 779–782
13. Lloyd, A.L., May, R.M.: Spatial heterogeneity in epidemic models. J Theor Biol **179**(1) (Mar 1996) 1–11
14. Keeling, M.J., Rohani, P.: Estimating spatial coupling in epidemiological systems: a mechanistic approach. Ecology Letters **5** (2002) 20–29
15. Anderson, R.M., May, R.M.: Infectious diseases of humans : dynamics and control. Oxford University Press (1991)
16. Brown, D.H., Bolker, B.M.: The effects of disease dispersal and host clustering on the epidemic threshold in plants. Bull Math Biol **66**(2) (Mar 2004) 341–371
17. Colizza, V., Barrat, A., Barthélemy, M., Vespignani, A.: The role of the airline transportation network in the prediction and predictability of global epidemics. Proc Natl Acad Sci U S A **103**(7) (Feb 2006) 2015–2020
18. Kermack, W.O., McKendrick, A.G.: A contribution to the mathematical theory of epidemics. Proc. Royal Soc. London **115**(1-2) (1927) 700–721

K-Ants Clustering - A New Strategy Based on Ant Clustering

Zahra Sadeghi[1], Mohammad Teshnehlab[2] and Mir Mohsen Pedram[3]

[1]Computer Engineering Department, Science and Research Branch, Islamic Azad University - Member of Young Researchers Club
[2]Faculty member, Electronic Engineering Department, KNT University, Tehran, Iran
[3]Faculty member, Electronic Engineering Department, Tarbiat-Moallem University, Tehran, Iran
[1]z.sadeghi@seiau.ir, [2]teshnehlab@eetd.kntu.ac.ir, [3]pedram@tmu.ac.ir

Abstract. In this article a new method of clustering with artificial ants is proposed. Unlike conventional ant clustering algorithms the true number of clusters must be provided for this algorithm in advance. Clustering is done using groups of ants which are as many as the number of clusters. The goal of each group is to collect members of one cluster. Two new functions are defined inspired from existing pick up and drop down probability functions which are used for inserting and removing loads of ants. Experimental results of the method demonstrated better accuracy in comparison to k-means and ant based clustering algorithm.
Keywords: ant clustering, k-means algorithm

1 Introduction

Clustering is an important technique that has been studied in various fields with many applications such as image processing, marketing, data mining and information retrieval. The goal of clustering is partitioning data into some groups so as all the members of one group have a close relationship to each other. This relationship is often expressed as similarity/dissimilarity measurement and is calculated through distance function. Recently, algorithms inspired by nature are used for clustering. The swarm intelligence clustering models and algorithms have advantages in many aspects, such as self-organization, flexibility, robustness, no need of priori information, and decentralization [1]. Ant colony optimization (ACO) is a relatively new and expanding branch of intelligent systems. A feature of these algorithms is that the clustering objective is implicitly defined. Implicit here means that neither the overall objective of the process (i.e., clustering) nor the types of clusters sought are defined explicitly at any point during the clustering process [2]. Some previous works done on ant based clustering include adaptive setting of some parameters [3-6], use of pheromone traces for directing ants to direct ant movement towards promising grid positions[6-7] and the replacement of picking and dropping probabilities by fuzzy rules[8-9],[2].

In this paper, we introduce a new approach for clustering with ants in which the number of clusters must be provided for it in advance. The clustering is done using a square grid. Each ant has a load list that must be filled with the members of one cluster. So every ant is supposed to search for one distinct cluster. Ants use inserting and removing functions for adding a new member to their load list and to remove the wrong members from it. These functions are the modified form of the popular picking up and dropping down functions and we will introduce them in section 3.

2 The General Principle of Ant Clustering

Ant based clustering is a distributed process. The pioneers of this work are Deneubourg et al. [10]. Their model is known as basic model (BM) in which ants are simulated by simple agents that randomly move in an environment which is a square grid. Initially, each data object that represents a multi-dimensional pattern is randomly distributed over the 2-D space. Data items that are scattered within this environment can be picked up, transported and dropped by the agents in a probabilistic way. The picking and dropping operations are influenced by the similarity and density of the data items within the ant's local neighborhood. Deneubourg *et al*. proposed a computational model for spatial sorting. Their model is based on the following probability of picking and dropping functions:

$$Pp = (\frac{Kp}{Kp + f})^2 \tag{1}$$

$$Pd = (\frac{f}{Kd + f})^2 \tag{2}$$

Where Pp is the probability of picking, Pd is the probability of dropping, and Kp and Kd are constants. f is the perceived fraction of items in the neighborhood of the ant. The probability of picking (Pp) is increased if a data object is surrounded by dissimilar data, or when there’s no data in its neighborhood. Also, ants trend to drop data in the vicinity of similar ones, so the probability of dropping (Pd) is increased if ants are surrounded with similar data.

Deneubourg et al.’s model was later extended by Lumer et al. [11] to allow its application to exploratory data analysis. They modified the probability functions as the followings:

$$Pp(i) = (\frac{Kp}{Kp + f(i)})^2 \tag{3}$$

$$Pd(i) = \begin{cases} 2f(i) & if \;\; f(i) < Kd \\ 1 & if \;\; f(i) \geq Kd \end{cases} \tag{4}$$

Where Pp and Pd are as before, i.e., the probability of dropping and picking. In addition, they introduced the density function as the following:

$$f(i) = \max\left(0.0, \frac{1}{\sigma^2}\sum_{j \in L}(1 - \frac{\delta(i,j)}{\alpha})\right) \quad (5)$$

Where $\delta(i,j) \in [0,1]$ is the dissimilarity function between two data object i and j. $\alpha \in [0,1]$ is the scaling parameter that permits further adjustments of resulting values. σ^2 is the size of local neighbourhood L around the ant's current position. The basic ant clustering algorithm is like the one summarized as bellow [2], [12]:

Let $R \in [0,1]$ be a random number drawn for each use, the basic algorithm is then given by:

- Randomly scatter data items on a square grid.
- For a number of steps or until a criterion is met, repeat for each ant:
  - If the ant is unladen and placed on location *l* occupied by object *o*, the object is picked up if $R \leq P_{pick\_up}(o)$.
  - If the ant is carrying object *o* and placed on an empty location *l*, the object is dropped if $R \leq P_{drop\_down}(o)$.
  - Move the ant to a new randomly selected (neighbouring) location.

3 The General Idea of K-ants Clustering

Like the basic model our method uses grid for clustering but in contrast to that it doesn't make clusters on the grid. One major problem with the basic model of ant based clustering is that if there were no clear boundary between data objects on the grid then retrieving clusters can be done with ambiguity. Our proposed method uses k ants for finding k clusters and they collect data objects in their load list, so there is no need to the retrieving process. We have considered a grid with $s * s$ cells. The number of cells must be greater than the number of all objects. In the beginning of clustering, we generate ants as many as the number of the clusters. Each ant must find the members of one cluster. All ants move with different speeds on the grid. We have considered a maximum speed for all ants which is set as $max\_speed = \frac{s}{3}$. The step of movement of each ant is generated randomly between 1 and max_speed before each movement. We decrease the max_speed by one every 200 iterations. The density function (6) which was proposed by Handle et al. [13], has been used in this study.

$$f(i) = \begin{cases} \max\left(0.0, \frac{1}{\sigma^2}\sum_{j \in L}(1 - \frac{\delta(i,j)}{\alpha})\right) & if \quad \left(1 - \frac{\delta(i,j)}{\alpha}\right) > 0 \\ 0 & otherwise \end{cases} \quad (6)$$

Where $\delta(i,j) \in [0,1]$ is the dissimilarity function between two data object i and j. $\alpha \in [0,1]$ is a scaling parameter that permits further adjustments of resulting values. L is the load list of the ant that wants to insert a new item or remove an existing one from it. σ is the size of L. We have also used the probability of picking up and dropping down functions proposed by Lumer & Faieta in the modified form of inserting and removing which are defined according to formula (7) and (8), respectively. The value of the parameters that we have chosen for our experiment is presented in Table 1. These settings are selected because we got satisfying results using them.

$$P_{insert}(i) = \begin{cases} \left(\dfrac{f(i)}{K_{ins} + f(i)}\right)^2 & if\ f(i) < K_{ins} \\ 1 & if\ f(i) \geq K_{ins} \end{cases} \tag{7}$$

$$P_{remove}(i) = \begin{cases} 0 & if\ \ f(i) \geq K_{rem} \\ \left(\dfrac{K_{rem}}{K_{rem} + f(i)}\right)^2 & if\ \ f(i) < K_{rem} \end{cases} \tag{8}$$

Where K_{ins} and K_{rem} are the inserting and removing constants respectively.

Whenever an ant is located in an occupied cell, it tries to pick up the item and insert it into its load list according to inserting formula. Similarly, if an ant is located in an empty cell, it searches its load list and tries to remove one of its loads according to the removing formula. For the sake of accuracy, in every iteration each ant tries to merge its loads with other ants' load list. For this, each ant checks all of its loads for inserting into other ants' load list using the inserting probability function. In this way the wrongly clustered data can be returned back into their correct clusters and the error of clustering decreases. This is a very important process which prevents ants from falling into local minimum.

Table 1. Parameters of the density function and their value.

Parameter	***Value***
K_{ins}	0.15
K_{rem}	0.15
α	0.5

4 The Basic K-ants Clustering Algorithm

The detail of our proposed k-ants clustering algorithm is explained in Algorithm 1.

Algorithm 1. Basic form of proposed K-ants algorithm.

```
// Initialization part
Consider the number of clusters and generate equal ants
Spread the data and ants randomly on the grid and give all ants a random load.

// main
While iter < max_iter
For i=1 to #ants
a = Choose a random ant
step = Generate a random speed from interval (1,max_speed)
new_location=Move (a, step)
If there is a load l in new_location then
Calculate the density function of l according to formula (6). L is the load list
```

```
of ant a
inserting_p = Calculate the inserting probability of load l according to formula
(7)
r = Generate a random number from interval (0, 1).
    If inserting_p > r then
        Enter the load l into the load list of ant a.
    End If
Else If there is no load in new_loacation then
For all loads l of ant a
Calculate the density function of l according to formula (6). L is the load list
of ant a
removing_p = calculate the removing probability of load l according to
formula (8)
r = Generate a random number from interval (0,1)
    If removing_p>r then
        remove load l from the load list of ant a
    End If
End For
End If

// update the load list of ant a
For k=1 to #ants
For l=1 to #load_list(a)
If k~=a then
f = Calculate the density function of l according to formula (6). L is the load
list of ant k
inserting_p = calculate the inserting probability of l according to formula (7)
r = Generate a random number from the interval (0,1)
    If inserting_p>r then
        insert the load l into load list of ant k and remove it from load list of
ant a
    End
End If
End For
End For
End For
End while
```

5 Algorithm Modifications

In order to improve the convergence speed and accuracy we have decided some modifications which are discussed in the following subsections.

5.1 Helper Ants

The grid space can be explored very slowly if the number of clusters is much less than the size of the dataset. To avoid this to happen, more than one ant can be used for gathering the members of one cluster. We have called them helper ants. By using helper ants clustering can be done much more quickly because of the cooperation of a number of ants for finding the members of one cluster. In this way every ant must always be aware of all the members of other ants of its group. Each time an ant wants to insert or remove an item it must take other members of its group into account. So the parameter L in formula (6) will change to the members of all load lists of the ants of one group i.e. all helper ants of one group. Note that an ant can only update its load list with an ant of opposite group. We have used 2 helper ants for each group in our implementation.

5.2 Blocking the Empty Cells

After a few iterations, most cells become empty. Stepping on the empty cells can only cause ants to put their loads on them. These loads must then be taken by other ants. This process is too time consuming and prolongs the gathering of members of clusters. To prevent this to happen, after some constant number of iterations, the empty cells become block. No ants can enter a block cell.

5.3 Stopping the Action of Putting on the Grid

Instead of putting down the loads iteratively and then picking them up, we have forbidden the action of putting on the grid after some constant number of iterations. In this way the correction of clusters can only be done by updating the load lists of ants. By this heuristic, the transfer of one load from one ant's load list to another can be done directly and the redundant action of putting on the grid will be avoided.

6 Experimental Results and Comparison

We have used the F-Measure [7] and Rand Index [8] for evaluating the clusters generated by our algorithm. The definitions of these functions can be found in [9], [10].

The datasets we used are summarized in Table 2 and the results of applying the evaluation functions (F-Measure and Rand Index) on them are shown in Table 3. It shows means and standard deviations (in parentheses) which is obtained over 30 independent runs with different initializations for the two measures. Additionally, Table 3 shows the error of clustering with respect to the number of wrongly clustered data. In all cases we have compared our results with k-means and ant-based clustering algorithm using LF model.

Table 2. Summary of the used data sets. dim is the dimensionality, k gives the number of clusters, and size shows the number of members of a cluster.

Name-k-size	Dim	***Source***
Iris-3*50	4	UCI
Square5-4*100	2	N([0,0],[2,2]),([0,5],[2,2]),([5,0],[2,2]),([5,5],[2,2])

Table 3. Test results on Iris dataset. Underlined bold face indicates the best and bold face the second best result out of the three algorithms.

Iris dataset	K-means	LF	Proposed Method
Min Error	8	**7**	**<u>5</u>**
Max Error	**19**	**<u>16</u>**	21
Average Error	12.18182	**9.85**	**<u>9.61905</u>**
F-Measure	**0.86665** (0.036583)	0.825911 (0.0148461)	**<u>0.918927</u>** (0.030938)
Rand Index	0.854525 (0.030061)	**0.855422** (0.01814619)	**<u>0.905753</u>** (0.030487)
Total Iteration	**<u>100</u>**	400000	**5000**
Square5	K-means	LF	Proposed
Min Error	**<u>1</u>**	5	**2**
Max Error	24	**10**	**12**
Average Error	8.47	**6.22**	**<u>4.34</u>**
F-Measure	**0.832526** (0.040423)	0.823462 (0.0411472)	**<u>0.862134</u>** (0.022634)
Rand Index	**0.844524** (0.042345)	0.835422 (0.02814619)	**<u>0.875753</u>** (0.020487)
Total Iteration	**<u>100</u>**	400000	**5000**

The results show that unlike k-means algorithm, our proposed algorithm doesn't trap into local minimum and its results are independent to the initialization values. In addition, the clusters obtained from the proposed algorithm are significantly better than k-means and ant clustering algorithm in term of accuracy.

7 Conclusion

In this paper, we presented a new strategy for clustering using artificial ants in which groups of ants tries to do clustering by inserting and removing operations. The number of groups of ants is equal to the number of clusters which must be specified in prior to clustering. All data objects and ants are spread randomly on the grid. Each ant contains a load list which is initialized with a random object at first. Ants search the grid and try to collect similar data to their load. The load list of ants of each group constructs each cluster and so in contrast to the basic ant clustering algorithms there is no need for the extra process of cluster retrieval from grid. The experimental results dedicated better performance in comparison to k-means algorithm and the LF model of ant clustering. It also outperforms k-means algorithm because it doesn't fall in local minima and its results are not dependant to the initialization part of algorithm.

References

1. Bin, W. and Zhongzhi, S.: A Clustering Algorithm Based on Swarm Intelligence. Proc. of the International Conferences on Info-tech and Info-net, Beijing, China, pp. 58-66 (2001)
2. Handl, J. and Meyer, B.: Ant-based and Swarm-based Clustering. Swarm Intelligence, pp. 95-113 (2007)
3. Deneubourg, J.-L., Goss, S., Franks, N., Detrain, C., and Chretien, L.: The Dynamics of Collective Sorting: Robot-Like Ant and Ant-Like Robot. In From Animals to Animats: Proc. of the First Int. Conf. on Simulation of Adaptive Behavior, Meyer, J.-A. & Wilson, S., Editors, Cambridge: MIT Press, pp. 356-365 (1991)
4. Lumer, E. and Faieta, B.: Diversity and Adaptation in Populations of Clustering Ants. In From Animals to Animats 3: Proceedings of the Third Int. Conf. on Simulation of Adaptive Behavior, Cliff, D., Husbands, P., Meyer, J.-A. and Wilson, S. W., Editors, Cambridge: MIT Press, pp. 501-508 (1994)
5. Magnus, E. and Pedersen, H.: Ant Colony Clustering & Sorting (2003)
6. Handl, J., Knowles, J. and Dorigo, M.: Strategies for the Increased Robustness of Antbased Clustering. Self-Organising Applications: Issues, Challenges and Trends Lecture Notes in Computer Science, **2977**, Springer-Verlag. pp. 90-104 (2003)
7. Rijsbergen, C. V.: Information Retrieval. 2nd edition, Butterworths, London, UK (1979)
8. Rand, W.: Objective Criteria for the Evaluation of Clustering Methods. Journal of American Statistical Association, **66**(336), pp. 846-850 (1971)
9. J. Handl.: Ant-Based Methods for Tasks of Clustering and Topographic Mapping: Extensions, Analysis and Comparison with Alternative Methods. Masters Thesis, Chair of Artificial Intelligence, University of Erlangen-Nuremberg, Germany (2003)
10. Handl, J., Knowles, J. and Dorigo, M.: On the Performance of Ant-Based Clustering. Design and Application of Hybrid Intelligent Systems. Frontiers in Artificial Intelligence and Applications, **104**, IOS Press, Amsterdam, The Netherlands, pp. 204-213 (2003)