TECHNICAL REPORT

Report No. CSAI2003-02
Date: November 2003

Proceedings of CSAW’03

Gordon J. Pace
Joseph Cordina
(editors)
Proceedings of CSAW’03

Gordon J. Pace
University of Malta
gordon.pace@um.edu.mt

Joseph Cordina
University of Malta
joseph.cordina@um.edu.mt

Abstract: This report contains the proceedings of the first Computer Science Annual Workshop (CSAW’03) — the research workshop held by the Department of Computer Science and AI of the University of Malta.
Preface

This volume contains the proceedings of the first Computer Science Annual Workshop (CSAW) organised by the Computer Science and AI department of the University of Malta. It should serve as an snapshot of the research currently taking place at the department by members of staff and graduate students.

Departmental workshops are an excellent means of getting everyone together to discuss their work — regularly leading to potential cooperation between different people and different research groups. Nowadays, most large CS departments hold such an annual workshop. However, the fact that at CS&AI we are a small department does not automatically imply better or easier research communication. In fact, lack of regular research presentations during the year (mainly due to the small number of researchers), means that one rarely knows exactly what the others are working on. It was thus felt necessary to organise CSAW.

The reasons for collecting short abstracts of the proceedings in a technical report are various. It is hoped that this will offer students embarking on a research degree, a first and early opportunity to write up their results and proposals. It should also showcase the research currently carried out in our department. As this report should witness, despite our small size our research still encompasses a wide and varied range of topics. Last but not least, it gives every one of us the opportunity to read what our fellow researchers are up to.

We would like to thank the CS&AI staff and research students for their cooperation, despite (or perhaps due to) our continuous nagging for presentation titles, reports, etc. Thanks also are due to the Malta Council for Science and Technology for providing us with the venue where CSAW is to be held.

Wishing you a research-wise fruitful year in anticipation of an even more interesting CSAW’04!

July 2003

Gordon Pace & Joseph Cordina
CSAW ’03 Organisers
# Table of Contents

Semantic Web Services Composition .................................................. 1  
*Charlie Abela*

Thue Systems for Pattern Recognition ............................................. 10  
*John Abela*

Implementing “The MATRIX” .......................................................... 18  
*Patrick Abela*

Ubiquitous Web Services .............................................................. 21  
*Malcolm Attard*

Mobile Positioning for Location Dependent Services in GSM Networks ........ 27  
*Josef Bajada*

Support Vector Machines ............................................................. 34  
*Steven Busuttil*

Grammatically Driven Class Derivation ........................................... 40  
*Ernest Cachia*

Maltese Speech Recognition over Mobile Telephony ............................. 45  
*Sinclair Calleja*

A Summary of Research in System Software and Concurrency at the University of Malta: I/O and Communication .............................................................. 48  
*Joseph Cordina*

Text-To-Speech Technologies for Mobile Telephony Services .................. 53  
*Nulphas-John Farrugia*

Open-domain Surface-Based Question Answering System ........................ 58  
*Aaron Galea*

Generic Chromosome Representation and Evaluation for Genetic Algorithms .......... 64  
*Kristian Guillaumier*
A Metrication and Comparison Framework for E-Commerce Systems .......................... 68
Mark Micallef

WWW2: Second Generation Web – Semantic Web ......................................................... 71
Matthew Montebello

A Log Analysis based Intrusion Detection System for the creation of a Specification Based Intrusion Prevention System ................................................................. 73
Andre’ Muscat

Enhancing Distributed Systems through Web Services .............................................. 80
Sergio Muscat

Computer-Aided Verification: How to Trust a Machine with Your Life ..................... 84
Gordon J. Pace

Finite State Analysis of Prepositional Phrases in Maltese ........................................... 92
Michael Rosner

A Risk Driven State Merging Algorithm for Learning DFAs ..................................... 99
Sandro Spina

Regulating Electronic Commerce using XML .............................................................. 103
Kenneth Spiteri

Achieving User Adaptivity in Hyperspace with HyperContext .................................. 106
Christopher Staff

Learning About the Learner: Deducing Learner Characteristics From Their Interaction with e-Learning Tools ................................................................. 110
Ingrid Talbot

A Summary of Research in System Software and Concurrency at the University of Malta: Multithreading ................................................................. 116
Kevin Vella

Author Index ............................................................................................................ 121
Semantic Web Services Composition

Charlie Abela
Department of Computer Science and AI, University of Malta

Abstract. Web services are becoming the most predominant paradigm for distributed computing and electronic business. They are self-contained Internet accessible applications that are capable not only of performing business activities on their own, but they also possess the ability to engage with other Web services in order to build new value-added services. Both academic and industrial bodies have been investigating issues regarding service descriptions, discovery and invocation, but automated service composition was somewhat neglected. The latter involves automated methods for constructing a sequence of Web services to achieve a desired goal. In this work we present initial research that focuses on the issue of automated service composition in conjunction with the Semantic Web. In this report we propose a composition engine that will automatically handle the integration of Web services through the use of a Web service description language such as DAML-S, the planning of workflow definitions, scheduling of tasks, status monitoring of the execution process, handling of faults and communication with other entities such as user agents, service registries and other composition engines.

1 Introduction

Web services are a relatively new technology for building distributed web applications. The three-level architecture for web services defined by Microsoft, IBM and Ariba, includes UDDI (Universal Discovery Description Integration) [1], WSDL (Web Services Description Language) [2] and SOAP (Simple Object Access Protocol) [3].

SOAP and WSDL are designed to provide descriptions of message transport mechanisms and for describing the interface used by each service. However, neither SOAP nor WSDL allow for the automatic location of web services on the basis of their capabilities. UDDI, on the other hand provides a registry of businesses and web services. UDDI describes businesses by their physical attributes such as name, address and services that they provide. In addition, UDDI descriptions are augmented by a set of attributes, called Tmodels, which describe additional features such as classification of services within taxonomies. Since UDDI does not represent service capabilities, it is of no help to a search for services based on what they provide.

A limitation that surrounds XML based standards, such as those mentioned above is their lack of explicit semantics by which two identical XML descriptions could mean totally different things, depending on the context in which they are used. This limits the capability of matching Web services. This is important because a requester for a Web service does not know which services are available at a certain point in time and so semantic knowledge would help in the identification of the most suitable service for a particular task.

The effort to integrate semantics into Web services started with the now standardized RDF and evolved with the creation of DAML+OIL and in particular with DAML-S and OWL-S (where S stands for Services) [4]. Neither of these are W3C standards, but the work done by these working groups is considered as being very important. Infact the WOWG (Web Ontology Working Group)
considered these languages as their initial step in the creation of the new Web Ontology Language called OWL. OWL is heading for standardization and the work involved the creation of two specific subsets of the language that can help implementers and language users.

As defined by the WOWG in [6], OWL Lite was designed for easy implementation and to provide users with a functional subset that will get them started in the use of OWL. OWL DL (where DL stands for “Description Logic”) was designed to support the existing Description Logic business segment and to provide a language subset that has desirable computational properties for reasoning systems. The complete OWL language (called OWL Full to distinguish it from the subsets) relaxes some of the constraints on OWL DL so as to make available features which may be of use to many database and knowledge representation systems, but which violate the constraints of Description Logic reasoners.

The rest of the paper is structured as follows. Section 2 will present some background on service composition languages, how they fit in the Web services architecture and a comparison between the major players in this area. In section 3 we define the types of compositions and how these are handled by referring to some related work on composition of Web services. Section 4 will describe our proposal for a service composition engine by considering all the phases in the composition cycle and give initial ideas how such an engine can be implemented. Finally in section 5 we discuss some open issues and conclude this paper.

2 Service Composition Languages

In spite of all the interest shown from both industrial bodies and academic institutions, several obstacles are preventing them from harnessing efficiently the Web services technology. The current Web service model, as depicted in Figure 1 below, enables service discovery dynamically, using markup languages for describing service properties, however it does not account for automatic integration of one service with another. Extensive work has been done in the area of service discovery and matchmaking as described in [7], [8] and [9]. However, the dynamics of service composition still remains one of the most challenging aspects for researchers in academia and industry. Several ongoing Industrial initiatives in the development of service markup languages such as BPEL4WS (Business Process Execution Language for Web Services, also called BPEL for short) [10], XLANG [11], WSFL (Web Services Flow Language) [12], WSCI (Web Services Choreography Interface) [13] are aimed at service composition. These, though, have resulted in solution providing frameworks, which are targeted towards proprietary application development environments. The languages may be syntactically sound, however they lack semantics and expressiveness. Nonetheless, automated service recognition, mechanized service composition and service negotiation are still amiss from them. Service Composition also remains one of the most important goals of the Semantic Web.
language has to be adequately expressive and it has to have a well-defined semantics and a robust formal model so to facilitate the automated composition of services through software agents. Recent efforts towards some kind of standardization of these technologies have resulted in the creation of the Web Services Choreography Working Group (WSC-WG) [15].

2.1 Comparison Of Existing Languages

We have analyzed and compared the three most important efforts regarding service composition, namely, DAML-S, BPEL and WSCI. A more detailed discussion can be found in [14].

To realize the automation of service composition on the Web, a language needs to have well-defined semantics along with syntactical constructs. Semantics help in defining reasoners for machine interpretation of service description. DAML-S with its base firmly rooted in Description Logics has well-established formal semantics. The process and profile model have been structured to enable intelligent agents to interpret the markup and reason about the composition. BPEL and WSCI do not expose any form of semantics and therefore do not facilitate the process of automated composition.

Expressiveness of a language is a collection of features that makes it easy to use, self-documenting and elegant. DAML-S, BPEL and WSCI are quite expressive with respect to process modeling constructs. DAML-S however offers an advantage over the two in its capability of expressing the pre-conditions and effects of service execution. Since DAML-S is an ontology, apart from XML data types, it also exposes a well-defined type system that enables reasoning about relationships between DAML-S classes. WSDL is restricted in its expressiveness of service behaviour to input/output as XML types. BPEL and WSCI, which derive WSDL port information for service description, therefore have limited expressivity in terms of typing mechanism.

Error handling and transaction management in case of service failure has to be an integral part of any composition model. Exception handling and Transaction constructs are present in both WSCI and BPEL but not in DAML-S. Fault and compensation handlers in BPEL seem to be more clearly defined then in WSCI. Both WSCI and BPEL allow roles to be defined. However no such construct is yet available in DAML-S. This is an important issue since roles help identify the responsibilities of partners in the composition. Correlation mechanism supported by WSCI and BPEL is important to synchronize the messages that are received by a service from different entities. This feature is currently not supported by DAML-S.

The process of marking up services using any of these languages is a cumbersome one if carried out manually. Editors and Engines are needed for creating, parsing and executing processes written
in these languages. The support for tools is very limited as language development is ongoing. An engine for BPEL is available at [6]. A number of efforts are in place for development of tools for DAML-S. A semi-automated service composer has been developed at University of Maryland. The SunONE WSCI Generator supports WSCI, however it does not provide means for testing the generated markup.

Extensibility of language constructs is necessary for enhancing the interface definitions. BPEL and WSCI allow for this with additional constructs from other XML namespaces. DAML-S allows this extensibility through the import construct and also through inheritance. Addition of rules is an important factor to allow for the different types of reasoning domains that are required by Web services. DAML-S is more at an advantage as regards the ease of incorporating rule definitions then the other languages since they are not based on a formal semantics. Security and Privacy issues have not been exclusively handled in any of these languages. They have been mentioned as future work, however currently the specifications of any of these languages do not provide mechanism to enforce security and privacy within the composition. Qualities of Service (QoS) [16] requirements are handled to some extent in DAML-S through its profile model, however there are no explicit QoS monitoring mechanisms available for BPEL and WSCI.

3 Web Service Composition

With the rapid expansion of Web services related applications in fields such as e-business, e-government and e-health, there is an increase in the demand for frameworks and infrastructures that can be used to develop applications that use Web service composition. In this section we first present a taxonomy of service composition types and then we refer to a number of initiatives that are trying to tackle the issue of composition, in particular, research on automated composition of Web services.

3.1 Service Composition Categories

Service composition as defined in [17] as the process through which newly customised services are created from existing ones by a process of dynamic discovery, integration and execution of those services in a deliberate order to satisfy user requirements. This process can be seen from two perspectives as described in [18]: the first is proactive versus reactive composition and the other is mandatory versus optional composition.

Proactive or static composition refers to offline or pre-compiled composition. Services that compose in such a manner are usually stable (i.e., they do not change very frequently) and are highly requested over the Web. Reactive or dynamic composition refers to the creation of services on the fly. Such composition requires some management facility to take responsibility of collaborating with the different sub-services to provide the composite service to the client. This interaction cannot be predefined and varies according to the dynamic situation. Reactive composition is better to exploit the present state of services and to provide certain runtime optimisations based on real-time parameters like bandwidth and cost of execution of the different sub-services.

Mandatory composite services refer to the situation where by all the sub-services must participate for the proper execution. These types of services are dependent on the successful execution of other services to produce the required result. Optional composite services are the opposite of the former and they do not necessarily need the participation of certain sub-services for the successful execution of a user query.
3.2 Related Work

There exists some work done on service composition that mainly focused on the dynamic execution of services. We also refer to ongoing research that is focused on automating this process.

The dynamic service composition called software hot swapping has been developed at the Carleton University, Canada [19]. This work is successor to the research in the field of dynamic software component upgrading at runtime. They have identified two different ways of carrying out heterogeneous service composition. The first method involves the formation of a composite service interface by which the necessary available services are exposed to a client. The advantage of this technique is the speed that a composite service can be created and exported. A second method creates a standalone composite service that is more suitable when the performance of the composite service is more critical. Such a service is created by dynamically assembling the available services by the way of pipes and filters, while all the service components remain independent.

eFlow [20] from HP labs, is an e-commerce services composition system. A composite service is modelled as a graph, which defines the order of execution among the nodes or different processes. The graph modelling a composite service consists of service nodes, event nodes or decision nodes. Service nodes represent simple or composite services. Event or decision nodes specify alternative rules that control the execution flow. Event nodes enable services and the process to receive various types of event notification. The eFlow engine offers the facility of being able to plug in new service offerings and enables adaptivity with several features such as dynamic service discovery, multi service nodes and generic nodes.

The Self-Serv [21] framework can compose Web services and the resulting composite service can be executed in a decentralised dynamic environment. The providers of the services participating in a composition collaborate in a peer-to-peer fashion to ensure that the control-flow dependencies expressed by the schema of the composite service are respected. A subset of statecharts has been adopted to express the control-flow perspective of the composite service. States can be simple or compound. The data-exchange perspective is implicitly handled by variables: which are the inputs and outputs, the parameters of services and events.

Golog [32] has been used for service composition by constructing general templates that are then modified based on user preferences, yielding a composite plan. The templates are not automatically built and constitute part of the plan.

We now turn our attention to some work that we are considering as being the initial idea for the composition engine we propose to implement. In particular we quote the work done in [22] which advocates a phased approach to service composition and which is collectively referred as the service composition life cycle. These phases describe the service composition process from the abstract specifications definition to their execution. Five phases are defined; the planning phase, the definition phase, the scheduling phase, the construction phase and finally the execution phase.

The planning phase assists the user in determining the series of operations that need to be retrieved and aggregated in order to satisfy the users request. Service requests define the desired service attributes and functionality, including temporal and non-temporal constraints between services, and the way the services are scheduled.

Once the services to be composed are chosen, an abstract definition of the composition is handled by the definition phase. This will provide for a customised specification of the new service that represents the definition or process model of the orchestrated service, together with grounding definitions that allow for the service bindings.

The scheduling phase is responsible for determining how and when the composed services will run and prepares them for execution. This phase is the first step towards a concrete definition
of the constructs defined in the process specification created in the previous phase. Scheduling includes the assessment of the service composability and conformance capabilities by correlation of messages and operations, and by synchronizing and prioritising the execution of the constituent services according to the process specification.

The construction phase results in the construction of a concrete and unambiguously defined composition of services that are ready to execute.

Lastly there is the execution phase that implements the service bindings and executes the services.

Though this work does not provide an automated composition process we think it is very important since it presents an interesting approach to the service composition problem. As described in [23], [24], [25] and [26] the semantic-web community is adapting AI planning techniques to give a solution to this problem. In AI planning, researchers investigate the problem of how to synthesise complex behaviours given an initial state. In the next section we propose our initial ideas for service composition based on the findings from the papers above together with some ideas of our own.

4 Composition Engine Some Issues

In this section we propose a composition engine that can be used by a user agent for the automated composition of Web services. In what follows we will give a rationale to the choices we made and discuss how some implementation issues will be handled.

We think that its important that Semantic Web technology is integrated with Web services as discussed in [27], where it is argued that to achieve the long term goal of seamless interoperability, Web services must embrace many of the representations and reasoning ideas proposed by the Semantic Web community and in particular the Semantic Web services community. We have tried to adopt this perspective to our work, and as discussed above we have looked in detail into a number of issues. First comes the composition language of choice. Though BPEL is somewhat more apt to fault and compensation handling, as described in our work [14], we think that it still requires work in the area of reasoning, since it is based on XML. In [27] BPEL is augmented with a semantic component to handle exactly this issue. Nonetheless we feel that DAML-S is somewhat more ideal at this early stage and hence we settled for this language for composition.

We are also opting to abide by the service composition cycle as defined in [22] and plan of adopting the same kind or architecture. This involves the creation of several modules mentioned in the paper, including, planner, definer, scheduler and executor. We will also be adding a reasoning and communications module.

As defined in the paper, the scheduler does not cater for fault and compensation handling nor for service execution monitoring. These are important issues that are missing from DAML-S as well and are discussed in [28]. If we pursue on including such functionality in our engine, then we must enhance the DAML-S language to handle these issues. This is not a trivial task and is still being debated. The other task handled by the scheduler is the decision of which service to execute and when. The scheduled service is then linked with the executor to be executed.

As regards the communication handler, the idea is to include the functionality that the engine can communicate with user agents or other composition engines, as the necessity might arise, hence creating a distributed environment rather than a centralised one. For this aim we plan to use JADE (Java Agent Development Framework)[29]. JADE is a software framework fully implemented in Java. It simplifies the implementation of multi-agent systems through a middle-ware that claims to comply with the FIPA (Foundation for Intelligent Agents) [30] specifications and through a set
of tools that supports the debugging and deployment phase. The agent platform can be distributed across machines (which not even need to share the same OS) and the configuration can be controlled via a remote GUI. The configuration can be even changed at run-time by moving agents from one machine to another one, as and when required. We also plan to use DAML+OIL as the ACL (agent communication language) that will be embedded into SOAP messages.

The reasoning module will involve the use of JTP (Java theorem Prover) [31]. This is an object-oriented modular reasoning system. JTP is based on a very simple and general reasoning architecture. The modular character of the architecture makes it easy to extend the system by adding new reasoning modules (reasoners), or by customizing or rearranging existing ones. The system is implemented in Java and this facilitates both extending JTP’s functionality and embedding JTP in other systems.

The definer will be required to generate new DAML-S process specifications from the individually composed service specifications.

The system will also require the facility of a matchmaker or UDDI registries from where the services can be retrieved and which can also be used to store the newly composed service definitions for reusability reasons. But this is an issue that we will be assuming as already in place hence it is out of scope of our work.

5 Discussion

At this stage we are reviewing several AI planning issues that can be used in the engine. We are trying to find a compromise between the work that already exists and the work that we might have to do to extend/adopt such research for our needs. Since this is a relatively new area we are faced with limited reviews and therefore require our own initiative to handle these tasks. For example
we might integrate the planner and scheduler into one module depending on the algorithm that we choose to adopt. Nonetheless we feel confident that since we already have the architecture in place then work can be initiated.

References

1. Universal Discovery Description and Integration Protocol: http://www.uddi.org/
2. Web Services Description Language http://www.w3.org/TR/wsdl
3. Simple Object Access Protocol http://www.w3.org/TR/soap12-part0/
4. DAML-S and OWL-S: http://www.daml.org/services/daml-s/0.9/
5. Web ontology Group http://www.w3.org/2001/sw/WebOnt/
6. OWL http://www.w3.org/TR/owl-ref/
16. Senthilandan Chandrasekeran, Composition, performance analysis and simulation of Web services, Masters Theses, University of Georgia Athens, 2002.

29. DAML-S: Semantic Markup for Web Services, http://www.daml.org/services/daml-s/0.9/-daml-s.html

30. JADE: http://sharon.cselt.it/projects/jade/

31. FIPA: http://www.fipa.org/

32. JTP: http://www.ksl.stanford.edu/software/JTP/
Thue Systems for Pattern Recognition

John Abela
Department of Computer Science and AI,
University of Malta

Abstract. This report presents a synoptic overview of Thue Systems. Thue Systems were introduced in the early 1900s by the Norwegian mathematician and logician Axel Thue. In this report the author suggests ways in which such systems can be used in pattern recognition.

1 Introduction

Many problems in pattern recognition and machine learning can, albeit not always in a straightforward manner, be reduced to the simple procedure of testing for formal language membership or to grammar induction. If patterns can be represented as strings, or for that matter tree or graphs, one can test for membership of a class simply by testing whether a given pattern, encoded as a string, is a member of the language consisting of all string encodings of the patterns in a class. Applications in pattern recognition that fall under this category include speech recognition, ECG signals, sonar signal, OCR, ECG patterns, picture languages, and many others [6, 7]. Learning in this case becomes simply a case of grammar induction. Learning grammars, for reasons we shall not discuss here, is by no means an easy task [5]. Some formal languages can be described by a set of re-writing rules that play the role of a grammar, i.e. they can be used to specify, generate, or recognize the strings in the language. Before proceeding any further we present some simple definitions.

2 Reduction Systems

The Norwegian mathematician and logician Axel Thue [12] considered the following problem: Suppose one is given a set of objects and a set of rules (or transformations) that when applied to a given object yield another object. Now suppose one is given two objects $x$ and $y$. Can $x$ be transformed into $y$? Is there perhaps another object $z$ such that both $x$ and $y$ can be transformed into $z$?

This problem became known as the word problem. Thue published some preliminary results about strings over a finite alphabet. Although Thue restricted his attention to strings he did suggest, however, that one might be able to generalize this approach to more structured combinatorial objects such as trees, graphs, and other structured objects. This generalization was later developed and the result was reduction systems. Reduction systems are so called because they describe, in an abstract way, how objects are transformed into other objects that are, by some criterion, simpler or more general. Reduction systems fall under the general name of replacement systems [8]. Replacement systems are an important area of research in computer science and have applications in automated deduction, computer algebra, formal language theory, symbolic computation, theorem proving, program optimization, and machine learning.

The reader is referred to [3, Chapter 1] for an exposition.
Definition 21 (Reduction System) [3, page 10]
Let $S$ be a set and let $\to$ be a binary relation on $S$. Then:

1. The structure $R = (S, \to)$ is called a reduction system. The relation $\to$ is called the reduction relation. For any $x, y \in S$, if $x \to y$ then we say that $x$ reduces to $y$.
2. If $x \in S$ and there exists no $y \in S$ such that $x \to y$, then $x$ is called irreducible. The set of all elements of $S$ that are irreducible with respect to $\to$ is denoted by $\text{IRR}(R)$.
3. For any $x, y \in S$, if $x \xrightarrow{*} y$ and $y$ is irreducible, then we say that $y$ is a normal form of $x$. Recall that $\xrightarrow{*}$ is the reflexive, symmetric, and transitive closure of $\to$.
4. For any $x \in S$, we denote by $\downarrow_R (x) \overset{\text{def}}{=} \{ y \in S \mid x \xrightarrow{*} y, y \text{ is irreducible} \}$, the set of normal forms of $x$ modulo $R$.
5. If $x, y \in S$ and $x \to y$, then $x$ is an ancestor of $y$ and $y$ is a descendant of $x$. If $x \to y$ then $x$ is a direct ancestor of $y$ and $y$ is a direct descendant of $x$.
6. If $x, y \in S$ and $x \not\to y$ then $x$ and $y$ are said to be equivalent. □

Notation 22 (Reduction System) [3, page 10]
Let $R = (S, \to)$ be a reduction system. Then:

1. For each $x \in S$:
   Let $\Delta(x)$ denote the set of direct descendants of $x$ with respect to $\to$. Thus, $\Delta(x) \overset{\text{def}}{=} \{ y \mid x \xrightarrow{\ast} y \}$. Also, let $\Delta^{\pm}(x) \overset{\text{def}}{=} \{ y \mid x \xrightarrow{\pm} y \}$ and $\Delta^{\ast}(x) \overset{\text{def}}{=} \{ y \mid x \xrightarrow{\ast} y \}$. Thus, $\Delta^{\ast}(x)$ is the set of descendants of $x$ modulo $\to$.
2. For each $A \subseteq S$:
   Let $\Delta(A)$ denote the set of direct descendants of $A$ with respect to $\to$. Thus, $\Delta(A) = \bigcup_{x \in A} \Delta(x)$. Also, let $\Delta^{\pm}(A) \overset{\text{def}}{=} \bigcup_{x \in A} \Delta^{\pm}(x)$ and $\Delta^{\ast}(A) \overset{\text{def}}{=} \bigcup_{x \in A} \Delta^{\ast}(x)$ Thus, $\Delta^{\ast}(A)$ is the set of descendants of the subset $A$ modulo $\to$.
3. For each $x \in S$:
   Let $\nabla(x)$ denote the set of direct ancestors of $x$ with respect to $\to$. Thus, $\nabla(x) \overset{\text{def}}{=} \{ y \mid x \xleftarrow{\ast} y \}$. Also, let $\nabla^{\pm}(x) \overset{\text{def}}{=} \{ y \mid x \xleftarrow{\pm} y \}$ and $\nabla^{\ast}(x) \overset{\text{def}}{=} \{ y \mid x \xleftarrow{\ast} y \}$. Thus, $\nabla^{\ast}(x)$ is the set of ancestors of $x$ modulo $\to$.
4. For each $A \subseteq S$:
   Let $\nabla(A)$ denote the set of direct ancestors of $A$ with respect to $\to$. Thus, $\nabla(A) \overset{\text{def}}{=} \bigcup_{x \in A} \nabla(x)$.
   Also, let $\nabla^{\pm}(A) \overset{\text{def}}{=} \bigcup_{x \in A} \nabla^{\pm}(x)$ and $\nabla^{\ast}(A) \overset{\text{def}}{=} \bigcup_{x \in A} \nabla^{\ast}(x)$ Thus, $\nabla^{\ast}(A)$ is the set of ancestors of the subset $A$ modulo $\to$.
5. Note that $\xrightarrow{\ast}$ is an equivalence relation on $S$. For each $s \in S$ we denote by $[s]_R$ the equivalence class of $s$ mod($R$). Formally, $[s]_R \overset{\text{def}}{=} \{ y \mid y \xrightarrow{\ast} x \}$. Also, for any $A \subseteq S$, $[A]_R \overset{\text{def}}{=} \bigcup_{x \in A} [x]_R$. □

Definition 23 [3, page 10]
Let $R$ be a reduction system.

1. The common ancestor problem is defined as follows:
   Instance: $x, y \in S$.
   Problem: Is there a $w \in S$ such that $w \xrightarrow{\ast} x$ and $w \xrightarrow{\ast} y$? In other words, do $x$ and $y$ have a common ancestor?
2. The common descendant problem is defined as follows:
   Instance: $x, y \in S$.
   Problem: Is there a $w \in S$ such that $x \xrightarrow{\ast} w$ and $y \xrightarrow{\ast} w$? In other words, do $x$ and $y$ have a common descendant?
3. The word problem is defined as follows:
   **Instance:** \( x, y \in S \).
   **Problem:** Are \( x \) and \( y \) equivalent under \( \leftrightarrow \)?

In general these problems are undecidable [3]. However, there are certain conditions that can be imposed on reduction systems in order for these questions to become decidable.

**Lemma 21** Let \((S, \rightarrow)\) be a reduction system such that for every \( x \in S \), \( x \) has a unique normal form. Then \( \forall x, y \in S \), \( x \leftrightarrow y \) if and only if the normal form of \( x \) is identical to the normal form of \( y \).

**Proof of Lemma** Let \( x, y \in S \) and let \( x' \) and \( y' \) denote the normal forms of \( x \) and \( y \) respectively.

\[ \Rightarrow \) Suppose that \( x \leftrightarrow y \) and \( x' \neq y' \). Then \( x \rightarrow^* y' \) since \( x \rightarrow^* y \) (by assumption) and \( y \rightarrow^* y' \) (by definition). Now \( y' \) is irreducible (by definition) and therefore \( x \) has two distinct normal forms: \( x' \) and \( y' \). This is a contradiction.

\[ \Leftarrow \) Suppose that \( x \) and \( y \) have a common normal form \( z \). Then, by definition, \( x \leftrightarrow z \) and \( y \leftrightarrow z \). The results follows from the symmetry and transitivity of \( \leftrightarrow \). \( \square \)

The proof of this lemma was omitted in [3]. The above result means that if for all \( x, y \in S \) we have an algorithm to check if \( x = y \) (very easy for strings), and also an algorithm to compute the unique normal forms of \( x \) and \( y \), then the word problem becomes always decidable.

**Definition 24** [3, page 11]
Let \( R \) be a reduction system.

1. \( R \) is **confluent** if \( \forall w, x, y \in S \), \( w \rightarrow^* x \) and \( w \rightarrow^* y \) implies that \( \exists z \in S \) such that \( x \rightarrow^* z \) and \( y \rightarrow^* z \).
2. \( R \) is **locally confluent** if \( \forall w, x, y \in S \), \( w \rightarrow x \) and \( w \rightarrow y \) implies that \( \exists z \in S \) such that \( x \rightarrow z \) and \( y \rightarrow z \).
3. \( R \) is **Church-Rosser** if \( \forall x, y \in S \), \( x \leftrightarrow y \) implies that \( \exists z \in S \) such that \( x \rightarrow z \) and \( y \rightarrow z \).

\( \square \)

**Definition 25** [3, page 12]
Let \( R \) be a reduction system. The relation \( \rightarrow \) is **noetherian** if there is no infinite sequence \( x_0, x_1, x_2, \cdots \in S \) such that \( x_i \rightarrow x_{i+1} \) for \( i \geq i \). If \( R \) is confluent and \( \rightarrow \) is noetherian then \( R \) is **convergent**.

---

**Fig. 1.** Properties of Reduction Systems.
If $R$ is a reduction system and $\rightarrow$ is noetherian then we are assured that at least one normal form exists. This means that the word problem and the common descendant problem are decidable. Furthermore, if $R$ is convergent then, for every $s \in S$, $[s]_R$ has one unique normal form. In addition, if $R$ is convergent then $R$ is confluent if and only if $R$ is locally confluent (see proof of Theorem 1.1.13 in [3]).

3 String-Rewriting Systems

A string-rewriting system, $T$, is simply a set of rewriting rules of the form $(l, r)$ where $l, r \in \Sigma^*$ for some finite alphabet $\Sigma$. The reduction system associated with $T$ is $R = (\Sigma^*, \rightarrow_T)$ where $\rightarrow_T$ is the reduction relation induced by $T$. If $(l, r) \in T$ implies that $(r, l) \in T$ then $T$ is called a Thue System otherwise it is called a semi-Thue System. In recent years there has been a resurgence of interest in Thue systems [3,4]. This interest is perhaps due to the advances made in computer algebra, automated deduction and symbolic computation in general [4]. There have also been a number of new results in the theory of replacement systems and this has spurred on more research. In machine learning we are concerned primarily with string-rewriting systems that induce reduction relations that are noetherian and, in particular, those that have only length-reducing rules, i.e. where $|l| > |r| \forall (l, r) \in T$. This property is desirable since it ensures that for any string $x \in \Sigma^*$, the normal forms of $x$ exist and are computable. One of the objectives of my research is to show how such string-rewriting systems can be used to (partially) specify certain interesting subclasses of formal languages.

3.1 Definitions and Notation

Definition 31 (String-Rewriting Systems) Let $\Sigma$ be a finite alphabet.

1. A string-rewriting system $T$ on $\Sigma$ is a subset of $\Sigma^* \times \Sigma^*$ where every pair $(l, r) \in T$ is called a rewrite rule.
2. The domain of $T$ is the set $\{l \in \Sigma^* | \exists r \in \Sigma^* \text{ and } (l, r) \in T\}$ and denoted by $\text{dom}(T)$. The range of $T$ is the set $\{r \in \Sigma^* | \exists l \in \Sigma^* \text{ and } (l, r) \in T\}$ and denoted by $\text{range}(T)$.
3. When $T$ is finite the size of $T$, which we denote by $|T|$, is defined to be the sum of the lengths of the strings in each pair in $T$. Formally, $|T| = \sum_{(l, r) \in T} (|l| + |r|)$.
4. The single-step reduction relation on $\Sigma^*$, $\rightarrow_T$, that is induced by $T$ is defined as follows: for any $x, y \in \Sigma^*$, $x \rightarrow_T y$ if and only if $\exists u, v \in \Sigma^*$ such that $x = uv$ and $y = uv$. In other words, $x \rightarrow_T y$ if and only if the string $y$ can be obtained from the string $x$ by replacing the factor $l$ in $x$ by $r$ to obtain $y$.

The reduction relation on $\Sigma^*$ induced by $T$, which we denote by $\rightarrow_T$, is the reflexive, transitive closure of $\rightarrow_T$.
5. $R_T = \{\Sigma^*, \rightarrow_T\}$ is the reduction system induced by $T$.
6. The Thue Congruence generated by $T$ is the relation $\equiv_T$ - i.e. the symmetric, reflexive, and transitive closure of $\rightarrow_T$. Any two strings $x, y \in \Sigma^*$ are congruent mod($T$) if $x \equiv_T y$.

For any string $w \in \Sigma^*$, the (possibly infinite) set $[w]_T$, i.e. the equivalence class of the string $w$ mod($T$), is called the congruence class of $w$ mod($T$).
7. Let $S$ and $T$ be two string-rewriting systems, $S$ and $T$ are called equivalent if they generate the same Thue congruence, i.e. if $\equiv_S = \equiv_T$.

Notes to Definition 31 For any string-rewriting system $T$ on $\Sigma$, the pair $(\Sigma, \rightarrow_T)$ is a reduction system. $T$ is a finite set of string pairs (rules) of the form $(l, r)$. Each rule can be interpreted to
mean ‘replace \( l \) by \( r \)’. The reduction relation induced by \( T, \rightarrow_T \), is usually much larger than \( T \) itself since it contains not just the rules of \( T \) but also all those strings pair \((x, y)\) such that, for some \( a, b \in \Sigma^* \), \( y = arb \) is obtained from \( x = alb \) by a single application of the rule \((l, r)\). In practice, for obvious reasons, \( \rightarrow_T \) is infinite.

Many of the properties of reduction systems we discussed in Section 2 apply also to \( R_T \). In particular, if \( T \) is a string-rewriting system on \( \Sigma \) and \( R_T = \{ \Sigma^*, \rightarrow_T \} \) is the reduction system induced by \( T \), then, for any two strings \( x, y \in \Sigma^* \):

- \( \rightarrow_T \) is confluent if \( w \stackrel{*}{\rightarrow}_T x \) and \( w \stackrel{*}{\rightarrow}_T y \) for some \( w \in \Sigma^* \), then \( \exists z \in \Sigma^* \) such that \( z \stackrel{*}{\rightarrow}_T \) \( x \) and \( y \stackrel{*}{\rightarrow}_T z \). \( T \) is therefore confluent if whenever any 2 strings have a common ancestor they also have a common descendant.
- \( \rightarrow_T \) is Church-Rosser if \( x \leftrightarrow_T \) \( y \) then \( \exists z \in \Sigma^* \) such that \( z \stackrel{*}{\rightarrow}_T \) \( x \) and \( y \stackrel{*}{\rightarrow}_T z \). Informally, \( \rightarrow_T \) is Church-Rosser if any pair of equivalent strings has a common descendant.
- \( \rightarrow_T \) is locally confluent if \( w \rightarrow_T x \) and \( w \rightarrow_T y \) for some \( w \in \Sigma^* \), then \( \exists z \in \Sigma^* \) such that \( z \rightarrow_T \) \( x \) and \( y \rightarrow_T z \). In other words, \( \rightarrow_T \) is locally confluent whenever any two strings have a common direct ancestor they also have a common descendant.

As from this point onwards, purely in the interests of brevity and clarity, we shall omit the subscript \( T \) and simply use \( \rightarrow \), \( \rightarrow^* \), and \( \leftrightarrow \) instead of \( \rightarrow_T \), \( \rightarrow^*_T \), and \( \leftrightarrow_T \).

**Definition 32** (Orderings on \( \Sigma^* \)) Let \( \triangleright \) be a binary relation on \( \Sigma \).

1. If \( T \) is a string-rewriting system on \( \Sigma \), \( \triangleright \) is said to be compatible with \( T \) if \( l \triangleright r \) for each rule \((l, r) \in T \).
2. \( \triangleright \) is a strict partial ordering if it is irreflexive, anti-symmetric, and transitive.
3. If \( \triangleright \) is a strict partial ordering and if, \( \forall x, y \in \Sigma^* \), either \( x \triangleright y \), or \( y \triangleright x \), or \( x = y \), then \( \triangleright \) is a linear ordering.
4. \( \triangleright \) is admissible if, \( \forall x, y, a, b \in \Sigma^* \), \( x \triangleright y \) implies that \( axb \triangleright ayb \). In other words, left and right concatenation preserves the ordering.
5. \( \triangleright \) is called well-founded if it is a strict partial ordering and if there is no infinite chain \( x_0 \triangleright x_1 \triangleright x_2 \cdots \). If \( \triangleright \) is well-founded but also linear then it is a well-ordering. \( \square \)

**Notes to Definition 32** It turns out that if \( T \) is a string-rewriting system on \( \Sigma \) then \( \rightarrow_T \) is noetherian if and only if there exists an admissible well-founded partial ordering \( \triangleright \) on \( \Sigma^* \) that is compatible with \( T \). (Lemma 2.2.4 in [3]). This is useful because, for reasons we outlined previously, we want to consider only string-rewriting systems that are noetherian. For any string-rewriting system \( T \), in order to establish whether \( \rightarrow_T \) is noetherian we need only find (or construct) an admissible well-founded partial ordering that is compatible with \( T \). In our case we usually opt for the length-lexicographical ordering, i.e. where strings are ordered according to length first and then lexicographically.

Notice also that for any string-rewriting system \( T \), the set of direct descendants of a string \( x \in \Sigma^* \) modulo \( T \), \( \Delta(x) \), is finite. This is true even if \( \rightarrow_T \) is not noetherian and follows from the fact that any string \( x \in \Sigma^* \) has a finite number of substrings and therefore the rules in \( T \) can only be applied in a finite number of ways. On the other hand, if \( \rightarrow_T \) is noetherian, then \( \forall x \in \Sigma^* \), the set of all descendants of \( x \), \( \Delta^*(x) \), is finite. This follows by König’s Infinity Lemma.
Definition 33 (Normalized String-Rewriting Systems) Let $T$ be a string-rewriting system on $\Sigma$. $T$ is normalized if, for every rule $(l, r) \in T$,

1. $l \in \text{IRR}(T - \{(l, r)\})$, and
2. $r \in \text{IRR}(T)$.

□

Notes to Definition 33 Informally, $T$ is normalized if and only if, for each rule $(l, r)$ in $T$, the left-hand side $l$ can only be reduced by the rule $(l, r)$ itself and the right-hand side $r$ is irreducible. If $T$ is a string-rewriting system that is not normalized, i.e. it contains rules whose right-hand side that is reducible, there is a polynomial time algorithm that on input $T$ will output a string-rewriting system $T'$ such that $T'$ is normalized and equivalent to $T$ [3, Page 47]. Unless otherwise stated, all string-rewriting systems we will consider from now on are normalized.

3.2 Length-Reducing String-Rewriting Systems

A string-rewriting system $T$ is called length-reducing if $(l, r) \in T$ implies that $|l| > |r|$. In other words the left hand side of a rule is always longer than the right hand side. The obvious implication of this property is that when a rule is applied to a string $x$ the resulting string $x'$ is always strictly shorter than $x$. Recall that if $S$ is any string-rewriting system on $\Sigma$ then $\rightarrow_S$ is noetherian if and only if there exists an admissible well-founded partial ordering $\triangleright$ on $\Sigma^*$ that is compatible with $S$. Therefore, let $\triangleright$ be the length-lexicographical partial order on $\Sigma^*$. Since $l \triangleright r$ clearly holds $\forall (l, r) \in T$ and also since $\triangleright$ is admissible and well-founded then we can conclude $\rightarrow_T$ is noetherian. There are three particular types of length-reducing we shall use in this report. We now give the definitions.

Definition 34 (Monadic String-rewriting Systems)

Let $T$ be a string-rewriting system on $\Sigma$. $T$ is called monadic if $T$ is length-reducing and $|r| = 1$ or $r = \varepsilon$, $\forall (l, r) \in T$.

Definition 35 (Special String-rewriting Systems)

Let $T$ be a string-rewriting system on $\Sigma$. $T$ is called special if $T$ is length-reducing and $r = \varepsilon$, $\forall (l, r) \in T$.

Definition 36 (Trivial String-rewriting Systems)

Let $T$ be a string-rewriting system on $\Sigma$. $T$ is called trivial if $r = \varepsilon$, and $l = a, a \in \Sigma$, $\forall (l, r) \in T$.

□

3.3 Congruential Languages

We now informally investigate the possibility (and, in my talk, the feasibility) of using string-rewriting systems to define, and also test for membership of, formal languages.

Let $T$ be a string-rewriting system over an alphabet $\Sigma$. How can we use $T$ to define a proper language $L \subset \Sigma^*$? In other words, are there any interesting, non-trivial languages induced by $T$? Of course, we must define exactly what it means for a language $L$ to be induced by a string-rewriting system $T$. We examine some possibilities.
Let $L_1$ be the set of all irreducible strings modulo $T$, i.e. $L_1 = \text{IRR}(T)$.

Let $L_2$ be the union of all the equivalence classes with respect to $T$.

We observe that $L_1$ can be finite or infinite depending on $T$. It turns out that the set $\text{IRR}(T)$, i.e. the set of all irreducible strings modulo some string-rewriting system $T$, is a regular language and a finite state automaton that recognizes $L_1$ can be constructed in polynomial time from $T$ [3, Lemma 2.1.3, Page 37]. Whether such languages are useful or not is open to discussion but a review of the literature does not reveal any particular use. $L_2$, it turns out, is $\Sigma^*$ itself. It appears, therefore, that $T$ by itself is rather limited for the purpose of defining formal languages. Suppose, however, that we use $T$ together with a finite number of equivalence (congruency) classes modulo $T$.

Definition 37 (Congruential Languages) [9]

Let $T$ be a finite string-rewriting system on $\Sigma$. A congruential language is any finite union, $C$, of congruency classes of $T$.

We specify a congruential language by the pair $(T, C)$. Since both $T$ and $C$ are finite sets, we have a finite description of the language. Congruential languages have been subject to study by various researchers [4, 9]. One interesting result is that all NTS languages are congruential [?]. A context-free grammar is said to be NTS if the set of sentential forms it generates is unchanged when the rules are used both ways. It is quite common, and also sensible, to restrict attention to congruential languages where $T$ has only length-reducing rules. This will ensure that $\rightarrow_T$ is noetherian and this guarantees that the normal forms for each string exist and are computable. We give below an example of congruential languages where $T$ has only length reducing rules.

Example 31 An Example of a Congruential Language

Let $\Sigma = \{a, b\}$ and let $T = \{(ab, \varepsilon), (\varepsilon, ab)\}$. This is the Dyck Language of matching parenthesis.

4 Thue Systems in Pattern Recognition

In my talk I will give examples of how Thue string rewriting system can be used in pattern recognition and also how learning algorithms can be developed to learn classes where the instances of the class are encoded as strings in some formal language. One of my objectives is to introduce Thue Systems to my colleagues especially those interested in formal language theory, compiling techniques, theoretical CS, and machine learning. The seminal work in this area is Ronald Book’s excellent exposition [3].

References


Implementing “The MATRIX”

Patrick Abela
Department of Computer Science and AI, University of Malta

Abstract. In this technical note we shall be considering how the asynchronous constructs of the Hoopla language can be combined with concurrency constructs of parallel languages to implement a framework which allows programmers to write programs which interact with one another in a non-deterministic way. This model draws inspiration from a recently released film, The Matrix Reloaded which proposes a fiction scenario where different programs are represented by real-life characters. We shall consider how the proposed model can be used to implement a fiction game based on The Matrix or used in a wider context to implement various asynchronous activities typically occurring on a distributed environment such as the web.

1 Synchronous constructs in Sequential and Concurrent Programming Languages

Consider a simple program which is to be executed sequentially:

```c
int x = 1;
int y = 2;
int z = 3;
int a = x+y;
int b = a+z;
```

Each instruction in the processor’s instruction set (assignment, conditionals etc.) is set to execute within a fixed time interval (of which duration depends on the system’s clock speed) such that it is safe for the processor to assume that as soon as a unit of time has passed it can proceed with the execution of the next instruction. Thus, the existence of a central clock mechanism and the guarantee that all instructions execute within a given unit of time allows for the sequential execution of our program. We shall be referring to this as synchronous computing in that each instruction is executed against a central clocking mechanism. Similarly the execution of concurrent programs is based on these same two requirements as the execution of a concurrent program involves essentially the execution of multiple sets of sequential instructions.

2 Asynchronous constructs in the Hoopla language

Hoopla is a language which was developed by the Hoopla research team at the Department of Computer Science at the University of Malta between 1997 and 1998. Hoopla programs are based on the premise that the underlying system upon which Hoopla programs are executed does not provide a central clocking mechanism and that instructions cannot be executed in a synchronous
manner. A virtual machine which simulates an asynchronous processor and which executes Hoopla programs was implemented as part of the research project.

Hoopla’s programming paradigm proposes programs as circuits of ‘molecular’ components which are reduced according to a set of pre-established reaction rules. Any group of interconnected molecules can be reduced if an applicable reaction rule is found. Figure 1 illustrates how a program with three interconnected molecular components is evaluated. Note that in this simple example the second addition cannot occur before the evaluation of the first addition. This is called ‘causal sequentiality’ and demonstrates one way of how sequential constructs can be implemented in such a paradigm.

The Hoopla language proposed a set of twelve molecules which could be used to express assignments, conditionals and all other constructs used for flow control. Most of the business logic used for the reduction of these circuits was implemented in the virtual machine itself such that new molecules could not be added easily. A revised version of the virtual machine was subsequently implemented such that the business logic for the interaction between the molecules was contained within the molecule components. These ‘micro-program molecules’ allowed the virtual machine to scale to support new molecule types. Various molecule sets were proposed and implemented on this virtual machine.

3 The Matrix: Reloaded

The recently released film, Matrix Reloaded presents an animated depiction of intelligent programs which act autonomously and which interact with one another. These programs interact and challenge one another through various techniques most of which are well known to programmers. For instance in one instance, a program clones itself repeatedly and attacks another program in what can be considered a typical ‘denial of service’ attack.

It is our idea to build a game whereby programmers could write their programs as molecules which are then launched to react asynchronously with other molecules released by other programmers. The game would be distributed on the web and the virtual machine would have the capability of reducing molecule circuits which span across a distributed environment. Reaction rules would be implemented by the programmers within the molecules.
4 Extending the Hoopla asynchronous model

A new model is being proposed to address the requirements of such an application. This model makes use of Hoopla’s asynchronous programming paradigm to have various components react with one another. Once two or more molecules are ready to react communication channels similar to those used in parallel languages are established such that information can be exchanged.

In order to make the game more realistic we might extend the model such that molecules or microprograms are associated with a position in a 3D co-ordinate system. Micro programs will react with one another if they share the same location. Using this notion it would be possible to build 3D defence systems which make use of ‘Wall’ molecules and ‘Door’ molecules. Similarly it would be also possible to implement special types of molecules which ‘spy’ on messages which are being sent as molecules between different sites. To counter balance this the programmers might need to implement special molecules which make use of keys to encrypt information.
Ubiquitous Web Services

Malcolm Attard
Department of Computer Science and AI,
University of Malta

1 Introduction

Ubiquitous coming from the Latin word *ubique*, means existing or being everywhere, especially at the same time. Web Services are loosely specified and coupled components distributed over the internet [23] with the purpose of being accessed and used ubiquitously by suppliers, customers, business and trading partners. This must be done independently of any tools or environment in use by any party involved. The basic service oriented architecture is based on the publishing of a service by a service provider, the location of a service by a service requestor and the interaction between the two based on the service description. The necessary functionality for the full adoption of such web services must include routing, reliable messaging, security, transactions, binary attachments, workflow, negotiation and management, web services description languages, choreography, orchestration and non-repudiation. A large number of companies and organizations are promoting this adoption and shifting their strategy to include this useful technology. A multitude of proposed standards and products have emerged in an attempt to meet the needs of this worldwide community of web services adopters. The core established standards include the Web Services Description Language (WSDL), the Simple Object Access Protocol (SOAP) and the Universal Description, Discovery and Integration (UDDI). The Web services Inspection Language (WSIL) is a more lightweight yet complimentary specification for service discovery[1]. Other definitions produced to tackle the required functions have not been fully standardized and many are still competing. For the needed functionality to be produced a number of related issues must be tackled. Here we look at some of the important ones, and how they are being tackled, we then shortly describe our proposed project and related works.

2 Transactions

Transactions are essential factor for web services adoption. In traditional scenarios the standard properties for a transaction are atomicity, consistency, isolation and durability (ACID). Now since most web services based applications are distributed remotely and usually owned by different parties, normal methods for transaction management are not as effective. Having central control over all the resources is very difficult in this scenario. The Business Transaction Protocol (BTP) specification aims to solve these problems by extending conventional methods to enable both ACID and non-ACID transactions using a two phase commit model based on structures defined as ATOMS and Cohesions [11]. The WS-Transactions and WS-Coordination specifications, recently released from WS-I, are specifications which like BTP aim to provide a mechanism where systems can inter-operate transactionally. The WS-Coordination framework defines services which include:

- An activation service
- A registration service
- A coordination service
while the WS-Transactions specification defines two type of protocols:

- atomic transactions protocols for short lived transactions
- business transactions protocols for long lived transactions

They provide the mechanism where transactions are described as series of activities. Services are created and registered via WS-Coordination services and their execution coordinated by the WS-Transaction protocols [5].

3 Security

A “user” may have a number of identities which need to be handled across different systems and services. The optimal approach would be to have a universal log-on, but this is currently impractical. Currently the leading initiatives in Web Services Security are the SAML, the Liberty Alliance Project based on SAML and WS-Security from WS-I [9]. SAML’s objective was to enable interoperability of security services across the internet, where security information is transmitted using XML. SAML provides a way to encapsulate the authentication process and provide transport for it. Thus the authority can determine what authentication to use. As discussed in [10] Microsoft have already attempted to use the Microsoft .NET Passport for universal single sign-on mechanism. Now WS-Security which is an XML and SOAP based message security model from IBM, Microsoft and Verisign has been recently submitted to OASIS. WS-Security extends the use of XML Encryption and XML Signature for protection and verification respectively. It has a higher level of abstraction than SAML and thus enables it to include SAML as a supported technology. Like SAML it does not specify authentication mechanisms but uses SOAP messages and describes how to attach signature and encryption headers to SOAP. The Liberty Alliance Project is focused towards a federated authentication framework where multiple identities can be linked together with the user’s consent. Liberty is based on three specifications:

- XML driven ways to communicate authentication information
- How these map to HTTP, SOAP and mobile protocols
- The authentication context

SAML is used as the assertion language for Liberty.

4 Quality of Service

Quality of service is another important factor for consideration when using web services. The following are major requirements [13] for a quality web service

- Availability - service needs to be present and ready for use.
- Integrity - maintain correctness of interaction.
- Accessibility - be capable of serving a web service request.
- Performance - have a certain level of throughput and latency.
- Reliability - maintain the service itself and the service quality.
- Regulatory - comply with standards and conform to the devised rules.
- Security - provide the agreed level of security.
- Transactional Qualities - conserve the stabilized transactional behavior.
WSDL does not specify semantics or aspects regarding the Quality of the Service. Thus QoS must be described in some other way. A lot of work has been done on QoS web based services at different levels including the network level, the system level, the web server level and the service level[22]. A service may be deployed to provide different service levels and assurances to different clients. Negotiation of web services and their QoS properties usually involves the creation of Service Level Agreements (SLAs). These must then by enforced, monitored and when expired, terminated as well. Third parties may also be involved in monitoring the QoS of a particular service [4].

5 Semantics

The Semantic Web is an extension of the current web in which information is given well-defined meaning [2]. It is based on RDF standards and driven by the W3C, together with a large movement of researchers and industrial partners. Its objective is have data defined and linked in such a way to achieve better use of the information on the internet. The DARPA Agent Mark-up Language has the purpose of marking up web pages such that they are given meaning, such that a DAML enabled browser or search engine may produce a better result than currently syntactically based ones. The DAML group of languages includes DAML-ONT,OIL, DAML+OIL and DAML-L. DAML+OIL[12] was produced from the convergence of the DAML-ONT, the first ontology language and OIL (Ontology Inference Layer) a logic based system. DAML-L is on the other hand is a complementary logical language which is able to express at least propositional Horn Clauses. These languages can be used to define the terms needed for the description of service invocation [8]. DAML-S is a proposed DAML+OIL ontology which has the purpose of describing the behavior, properties and capabilities of web services [15].

6 Composition

Web Services Composition involves the combination of a number of web services to produce a more complex and useful service. Choreography is the term used to define the tracking of message exchange between services while the term orchestration is used to refer to the services interaction involving the logic and order of interaction execution[17]. A number of composition languages have emerged to meet this purpose. These include IBM’s Web Services Flow Language (WSFL) and Microsoft’s XLANG whose concepts have been placed in the Business Process Execution Language for Web Services (BPEL4WS), a new specification from WS-I whose core members intuitively include Microsoft and IBM. This specification describes the modelling of web services behavior using business process interaction. The Web Services Choreography Interface (WSCI) is another specification produced by BEA, SAP and Intalio. It describes messages between the web services such that it defines the choreography as a exchange of messages. BPML produced by BPMI.org is yet another mark up language which takes the form of a meta language for describing business processes[25]. Using these representations and the relative composition engine one can compose web services as desired.

7 UbicWSCO

As the number of web services increase drastically with time and their use is extended to more common services, the task of effectively composing a web service will become overwhelming.
We aim to provide a detailed study of current web service composition definitions and methodologies and the development of a method, accompanied by its implementation (Ubiquitous Web Services Composer), which will enable us to compose a new service, from a number of other web services, automatically. This service will base its composition strategy on statistically based methods and user modelling where ‘end-user’ refers to either a company software business system, agent, or human user. Our approach will take into consideration pre-defined compositions, service usage history, the actual user needs, the services’ SLAs and also the usage context of the service. The ‘end-user’ will be offered a search like user interface which will upon request provide a number of plausible compositions which can then be deployed by the ‘end-user’. The selection criteria which will be used will also include transactional qualities and access options. We intend to use BPEL4WS together with the complementary specifications WS-Transactions and WS-Coordination based definitions for the purpose and to construct our system using Java driven IBM alphaworks technologies. We also intend to examine the application of such a mechanism and its interrelation with user facing web services [19].

8 Related Work

Web Services composition is a an active area of research and there exist a number of related frameworks which we will shortly outline here. The WeSCoS (Web Service Composition Framework) [14] was developed to provide a research framework for composition of loosely coupled services. It uses the XML programming language (XPL) for invoking services, iterating over results and creating output information in a document. In the ICARIS project [24] is based on Jini, JavaBeans and XML. It consists of two core management components, the Registration Manager and the Composition Manager, working on the Jini Infrastructure itself. The Composition Manager is responsible for the actual composition of general services while the Registration Manager is responsible for managing registration access rights. Composition is achieved using JavaBeans and the ERCSP (Extensible Runtime Containment and Services Protocol). The framework is generic and is not based on Web Services. Another automatic composition approach described in [26] is based on the semantic matching of web services parameters. METEOR-S, also from LSDIS is a follow up project to METEOR (Managing end-to-end operations) and provides MWSCF: (METEOR-S Web Service Composition Framework), a comprehensive framework for semantic web service composition. Within this framework the existing process composition techniques are enhanced by using templates to capture the semantic requirements of the process. In [16] another pattern composition approach is described. It involves the use of DAML-S subset, a situation calculus as first order
logic language for the description of changes based on named actions and petri nets for execution semantics. We also see the use of a petri net-based algebra to model control flows for web services composition in [7]. The QUEST framework [6] which is able to provide the best initial service composition path chosen using multiple QoS constraints and a dynamic re-composition mechanism to provide service path change on the fly, also based on QoS constraints and violations. SWORD is another tool set for Web Services Composition which uses a rule-based expert system to determine whether a service can be composed using existing services [18]. SWORD is focused towards information providing services and it can generate a functional composition plan given the functional requirements. HP eFlow supports specification, deployment and management of composite web services. A composite service in eFlow is described as a schema and modelled by a graph including service, decision and event nodes. To achieve adaptive service processes it provides a dynamic service discovery mechanism, multi-service nodes (allowing for multiple parallel activation of the same service node) and generic nodes (non statically bound services) [3]. Self-Serv is a framework based on state-chart modelling techniques where transitions are labelled using ECA rules. Thus services are composed in a declarative manner and the orchestration engine is based on a peer to peer model such that processing is distributed among the participants of the orchestration [21, 20].

References


Mobile Positioning for Location Dependent Services in GSM Networks

Josef Bajada

Department of Computer Science and AI,
University of Malta

Abstract. A feasible Mobile Positioning solution is often sought after by network operators and service providers alike. Location-dependent applications create a new domain of services which might not only be of interest to the next generation of mobile users but also create new potential revenue streams. Applications vary from emergency services and tracking to location-based information services, location-based billing and location-dependent advertising. Due to the shortcomings of location-related information present in GSM networks, and the lack of positioning functionality in most of the commonly sold mobile devices, a straightforward solution for mobile positioning does not currently exist. This research intends to propose cellular positioning methods which do not require any significant changes to the network or the mobile device itself, which are feasible and cost effective, and which provide sufficient accuracy for certain categories of location-based services. These techniques are based on the proper analysis of signal measurement data, probabilistic geometric computation of location areas likely to represent the user’s location, and the correlation of this data with information obtained from path loss models used in the design and planning of a mobile radio network.

1 Introduction

Being small, handy and increasingly utilitarian, the mobile phone has become the core device of mobile society. The addition of novel applications and capabilities make it even more personal and trusted, and thus a regular part of everyday life. The popularity of mobile phones and the number of mobile device users is continuously increasing, and at the same time mobile phone manufacturers are striving to introduce new feature-packed devices to hopefully attract potential new customers. These new features include device, internet and intranet connectivity. Consequently, consumers will gain access to content and services at any time and from virtually any geographic location. One must also not forget the emerging multimedia-enhanced services where the user experience is enhanced through images, audio and even video content. However, what makes mobile devices fundamentally different from other computers is their inherent mobility \[8\].

Location based services offer a huge number of possibilities for the definition of new services for 2G and 3G wireless networks. For some applications it is sufficient to determine the cell of the mobile terminal but other services such as emergency calls or navigation systems require a more accurate position determination framework.

Unfortunately, the GSM Network itself lacks positioning functionality since historically it was not designed to carry any location or telemetry information. One can consider the option of integrating with a separate positioning technology \[2\], but there are huge costs involved in upgrading a substantial part of the network’s base-stations with Location Measurement Units (LMUs) to be able to support technologies such as Enhanced Observed Time Difference (E-OTD) or Time of Arrival (TOA) \[9\]. Also, the limitations of technologies such as Global Positioning System (GPS)
and Assisted GPS (A-GPS) (for instance, the requirement of an open space for at least three satellites to be visible from the mobile terminal, the relatively long time for position fix, and the power required for the GPS receiver) make the technology inappropriate for commercial applications such as Localised Information Services. Apart from all this, technologies such as GPS or E-OTD are still missing from most common mobile handsets on the market.

Thus, the user’s location must be determined from data that is inherently present in the cellular network, which although not originally intended for mobile positioning, might be able to give enough clues to heuristically locate a mobile phone to an acceptable level of accuracy. This information consists of network parameters such as the Serving-Cell Identity, Timing Advance and Neighbouring Cell Measurements. This research presents and puts in practice cell-based location confinement and approximation algorithms that use techniques such as probabilistic geometry and path loss models. These algorithms are merged together to provide a positioning solution that can furnish a location dependent application with information pertaining the user’s position.

2 Probabilistic Geometric Elimination

In the interests of optimizing the efficiency of the network, the amount of spectrum that an operator has, as well as the quality of service for the user, it is necessary for the operator to know which Base Transceiver Station (BTS) is in closest proximity to each subscriber [10]. This is accomplished by handsets monitoring the signal strength and the Base Station Identification Code (BSIC) of the surrounding BTSs, to determine which BTS has the strongest signal, through which any calls or data communication will be routed.

Probabilistic Geometric Elimination groups the techniques that, from the provided network data, geometrically eliminate the areas that are highly improbable to resemble the subscriber’s location, and confine the regions that are likely to correspond to the obtained network measurements. Below is an overview of how this collected data contributes to obtain a probable subscriber location area.

2.1 Cell Identity

The simplest approach to cell-based localisation involves identifying the serving cell the mobile phone is using, since the coverage area of that BTS indicates the whereabouts of the user’s location. The coordinates of the BTS itself can be used as the first estimation of the mobile’s location. Often, most base station sites host more than one BTS and thus each BTS would cover a sector from the whole area around the site, therefore decreasing the probable subscriber location area to a specific sector. The achievable accuracy of the estimated location depends on the size of the cell, which might vary from a few hundred metres to several kilometers.

2.2 Timing Advance

Timing Advance (TA) is a crude measurement of the time required for the signal to travel from the MS to the BTS. In the GSM system, where each mobile station is allocated a specific frequency and time slot to send and receive data, this measurement is essential to make sure that time slot management is handled correctly and that the data bursts from the MS arrive at the BTS at the correct time (in the time slot allocated to them). The computed TA value is then used by the MS to advance transmission bursts so that the data arrives at the correct time slot. The resolution is
one GSM bit, which has the duration of 3.69 microseconds. Since this value is a measure of the round trip delay from the MS to the BTS, half the way would be 1.85 microseconds, which at the speed of light would be approximately equal to 550 meters. Thus, Timing Advance can give an indication of distance from the identified BTS, in steps of approximately 550 meters.

2.3 Network Measurement Reports

The Mobile Station continuously measures signal strengths from both the serving cell (the BTS it is attached to) and also its neighbouring cells. The serving cell supplies the MS with a list of adjacent cell frequencies which it should monitor and in return, the MS provides the results of up to six strongest signal strength measurements. When the signal strength of one of the adjacent cell frequencies is substantially higher than that of the serving cell, **Handover or Cell Reselection** takes place. The concept of Cell Reselection is similar to that of the Handover and is essentially the process of selecting another cell to attach to, with the difference that during Cell Reselection the MS is in idle mode while Handover occurs while the MS is engaged in a communication session.

The richer these Network Measurement Reports (NMR), the more clues are available for making a correct guess of the location of the mobile device. This information can be both useful but deceivingly difficult to be interpreted correctly so as to contribute to a greater accuracy over the initial location area obtained by using solely Cell-ID and TA. The received signal strength cannot be directly interpreted into the distance of the MS from the corresponding BTS since radio signals do not get attenuated by air with the same rate as the attenuation of other materials such as buildings and other solid obstructions. Therefore, direct triangulation or trilateration using signal strength is not possible, especially in urban areas.

**Fig. 1:** Deducing a probable location sector from three signal strength measurements originating from the same site.

However, NMR information can be used to deduce angular approximations. The fact that the signal from a particular antenna is being received with a greater strength than others can indicate that the bearing of the MS from the site is close to a certain degree to the azimuth that the antenna is facing. One can also quite confidently assume that signals originating from antennas on the same site (and with the same configuration) will be attenuated equally (since they will encounter the same obstructions and suffer the same path loss). For instance, a site located at coordinates \((x, y)\), which has the common three sector configuration, would typically be associated with three signal measurements \(a\), \(b\), and \(c\), which will contribute to the approximation of a probable location sector related that site, as shown in Figure 1. The constants \(J\) and \(K\) depend on antenna configuration parameters such as its radiated beam width.
By evaluating measurements from each neighbouring site present in the NMR, one can geometrically exclude more areas that are improbable to resemble the location of the MS, and thus the original area obtained from the Cell-ID (and possibly TA) would be cropped down to a smaller one by removing the parts that are highly unlikely to enclose the subscriber’s location. This principle will lead to complex geometric shapes which can be used as the initial search space for more complex and computationally intensive techniques that are to follow.

### 3 Path Loss Analysis

During the planning of wireless networks many path loss predictions for the whole coverage area of each BTS are computed in order to analyse and evaluate the coverage and interference scenario [4]. These path loss predictions are computed using propagation models which generate a 2D array representing the path loss for each pixel in the area around the BTS. The area represented by each pixel in this array depends on the desired granularity of the path loss predictions.

It is quite evident that the accuracy achieved from these models relies on the accuracy of the prediction models used. The three most common propagation models are discussed below.

**Hata-Okumura** This is an empirical model especially designed for the prediction of path loss in urban areas [6]. Obstacles, environmental features and local effects (shadowing or reflections) are not considered. Indoor coverage, especially in multi-floor buildings cannot be predicted with an acceptable accuracy.

**COST 231 Walfisch-Ikegami** This is an improved empirical and semi-deterministic model that was developed by the European COST research program [1]. This model gives particular attention to propagation over roof-tops (multiple diffractions in the vertical plane) and leads to higher accuracy in dense urban areas.

**Intelligent Ray Tracing (IRT)** The idea behind this deterministic model is to describe the wave propagation with rays launched from the transmitting antenna. These rays are reflected and diffracted at walls and similar obstacles [5]. Since propagation modelling through conventional Ray Tracing requires a high computational effort, IRT reduces this computation time by preprocessing building databases during which all geometric operations are pre-performed and stored. The main disadvantage with these kind of models is the high level of detail required to be present in the base map and building database.

Once the appropriate model is calibrated to predict to an acceptable accuracy, path loss predictions can be generated. A path loss prediction for each base station for each pixel is computed creating a 2D matrix representing the predicted signal strength of each location (up to a predefined granularity) for each BTS. The correct correlation of the actual measurements obtained from the mobile terminal with the predicted values will make localization of the mobile terminal possible.

#### 3.1 Dominant Cell Prediction

The path loss predictions discussed above can be used to identify the locations where a particular BTS is predicted to be the dominant cell, i.e. its frequency will have the strongest signal and thus it will be most likely selected to be the serving cell. This is often referred to as the best-server prediction. However, the best-server does not necessarily indicate all the locations where the BTS might be the serving cell since the handset might not perform cell reselection if the signal from the stronger cell does not exceed the current serving cell by a certain threshold, called cell reselection.
offset. Thus, we propose a variation of the best-server prediction which creates a new 2D matrix containing cell domination levels, indicating the difference in predicted signal strength between the particular cell and its strongest neighbour. Positive values in this array indicate the level by which the cell exceeds its strongest neighbour, while negative values indicate the level by which the cell is inferior (and therefore it is no longer the best server) to its strongest adjacent cell. Locations that are likely to be served by a particular cell $i$ are those where:

$$D_{i(x,y)} \geq 0 - C_i$$

with $D_{i(x,y)}$ being the cell domination level for cell $i$ at location $(x, y)$ and $C_i$ being the cell reselection offset for the BTS of cell $i$.

### 3.2 NMR Correlation

While dominant cell predictions help to confine the area where the mobile terminal is probably located, we still need to have a way to select the most probable location within that area. The NMR obtained from the mobile terminal can thus be compared to the predicted signal strengths at each location and the pixel that is predicted to match most to the obtained measurements will have the higher probability of being the actual location of the mobile terminal.

![Fig. 2: Correlation of NMR information with predicted signal strengths for a particular location.](image)

**Relative Signal-strength Correlation** The correlation method proposed by Wolfe et al. [5] involves performing a squared error analysis between the measured signal strengths and predicted signal strengths. This correlation method will produce good results outdoors for the generic case, but it will immediately suffer in urban environments where there are too much obstructions or the subscriber is indoors.

To make up for this problem, we propose a variant of this correlation method which basically involves the analysis of relative strength differences between the measured signal strength of the serving cell and the signal strengths of each neighbour compared with the predicted differences for the same sites. Thus, this variant will have an improved accuracy in locations where there is an approximately uniform signal drop due to situations such as the user being deep in a building.
Cell-placing Analysis The above correlation method does not take into consideration cells which were predicted to have a high signal strength but in fact had no measurement present in the NMR information. Thus, we propose another correlation method which analyses the cell’s placing (in the list of cells ordered by their signal strength) and not the actual signal strength. Thus, a cell which is present in the predictions but not in the NMR information will automatically inflict more error to the other cells and decrease the likelihood of that pixel to resemble the mobile terminal’s location.

Same-site Analysis Since two antennas on the same site are attenuated by the same obstructions, it can be assumed that the measured signal difference between two antennas residing on the same site will match closely to the predicted signal difference obtained from the path loss model. Thus, if more than one cell from the same site are present in the NMR information, a new probable location area, having the form of a narrow beam, can be generated from locations predicted to have similar signal strength differences between the same antennas. If the NMR information presents more than one site with this characteristic, then it would be also possible to triangulate the user’s location from beams originating from two sites.

4 Quality of Position

Apart from having a positioning system which locates mobile terminals with an acceptable accuracy, it is also important that the system is capable of determining the chances it has to guess the user’s position, or in other words, what would be the possible error associated with a particular location approximation [3]. This Quality of Position (QoP) would then be very useful for the LBS application, which would then be able to determine if the QoP is appropriate for the service and act accordingly. The methods for calculating the QoP we propose are based on analysing the distance from the approximated location to the vertices of the probable location area.

5 User Experience

Accurate positioning is only half of the story in delivering successful location-based services. One also has to focus on the service itself, the intuitiveness and user friendliness of the user interface, and the value it offers to the customer. There are also a number of customer oriented issues that must not be ignored, such as user-privacy [7].

Thus, it was felt that appropriate attention must also be given to the process of developing and delivering location-based services to the customer, to close the loop between mobile positioning technology and the actual user of the technology. This tight coupling between the user and the technology is especially important in our case since, the actual procurement of location information might have to be triggered or approved by the user himself through his mobile handset. This project also aims to present a prototype location service which abides with these guidelines and aims to provide the user with experience more than just content.

6 Conclusions

This research aims to present an extensive in-depth study of the value of Location Based Services and the technologies that can enable them in conventional GSM networks. A set of cell-based positioning techniques are proposed based on information available from the network and on accurate
path loss models. Part of the project also involves that these techniques are implemented and put on trial through the development of a prototype positioning system. The research will also put a special focus on the development of location based services themselves, user experience and other user-related issues.

References

Abstract. In this report we present an introductory overview of Support Vector Machines (SVMs). SVMs are supervised learning machines that can be analysed theoretically using concepts from computational learning theory while being able to achieve good performance when applied to real-world problems.

1 Introduction

The study of Support Vector Machines (SVMs) can be said to have been started by Vladimir Vapnik in the late seventies [15]. However it was only until the late nineties that the subject started to receive increasing attention [4, 7, 10].

Support Vector Machines, are supervised learning machines based on statistical learning theory that can be used for pattern recognition and regression. Statistical learning theory can identify rather precisely the factors that need to be taken into account to learn successfully certain simple types of algorithms, however, real-world applications usually need more complex models and algorithms (such as neural networks), that makes them much harder to analyse theoretically. SVMs can be seen as lying at the intersection of learning theory and practice. They construct models that are complex enough (containing a large class of neural networks for instance) and yet that are simple enough to be analysed mathematically. This is because an SVM can be seen as a linear algorithm in a high-dimensional space [13].

In this document, we will primarily concentrate on Support Vector Machines as used in pattern recognition. In the first section we will introduce pattern recognition and hyperplane classifiers, simple linear machines on which SVMs are based. We will then proceed to see how SVMs are able to go beyond the limitations of linear learning machines by introducing the kernel function, which paves the way to find a nonlinear decision function. Finally, we sum it all up and mention some areas in which Support Vector Machines have been applied and given excellent results.

2 Pattern Recognition and Hyperplane Classifiers

In pattern recognition we are given training data of the form

\[(x_1, y_1), \ldots, (x_\ell, y_\ell) \in \mathbb{R}^n \times \{+1, -1\},\]

that is \(n\)-dimensional patterns (vectors) \(x_i\) and their labels \(y_i\). A label with the value of +1 denotes that the vector is classified to class +1 and a label of -1 denotes that the vector is part of class -1. We thus try to find a function \(f(x) = y : \mathbb{R}^n \to \{+1, -1\}\) that apart from correctly classifying the patterns in the training data (a relatively simple task), correctly classifies unseen patterns too. This is called generalisation.
Statistical learning theory or VC (Vapnik-Chervonenkis) theory [16], shows that it is imperative that we restrict the class of functions that our machine can learn, otherwise learning the underlying function is impossible. It is for this reason that SVMs are based on the class of hyperplanes

$$\langle w \cdot x \rangle + b = 0; \ w \in \mathbb{R}^n, b \in \mathbb{R},$$

which basically divide the input space into two: one part containing vectors of the class $-1$ and the other containing those that are part of class $+1$ (see Figure 1). If there exists such a hyperplane, the data is said to be linearly separable (nonseparable otherwise). To find the class of a particular vector $x$, we use the following decision function

$$f(x) = \text{sign}(\langle w \cdot x \rangle + b).$$

2.1 The Optimal Hyperplane

As can be understood, there may be more than one hyperplane that correctly classifies the training examples (for instance, in Figure 1 the hyperplane could be closer to class $-1$). It has been shown that the hyperplane that guarantees the best generalisation performance is the one with the maximal margin of separation between the two classes [6, 7]. This type of hyperplane is known as the optimal or maximal margin hyperplane and is unique.

The optimal hyperplane can be constructed by solving a convex (no local minima, therefore any solution is global) optimisation problem that is minimising a quadratic function under linear inequality constraints. The solution to this problem has an expansion in terms of a subset of the training examples that lie on the margin, called support vectors (see Figure 2). Support vectors contain all the information needed about the classification problem, since even if all the other vectors are removed the solution will still be the same. The details of the calculations will be omitted but can be found in a number of our references (see for instance [6]).

Finally, another very important property of hyperplane classifiers that needs to be emphasised, is that both the optimisation problem (used to find the optimal hyperplane) and the decision function

![Fig. 1. A separating hyperplane ($w, b$) for a two dimensional (2D) training set.](image-url)
Fig. 2. A maximal margin hyperplane with its support vectors encircled.

(used for the actual classification of vectors) can be expressed in dual form which depend only on dot products between vectors. The dual representation of the decision function is

\[ f(x) = \text{sign} \left( \sum_{i=1}^{\ell} y_i \alpha_i \langle x \cdot x_i \rangle + b \right), \]

where \( \alpha_i \in \mathbb{R} \) is a real-valued variable that can be viewed as a measure of how much informational value \( x_i \) has. Thus for vectors that do not lie on the margin (i.e. non support vectors) this value will be zero.

### 3 Feature Spaces and Kernels

Linear learning machines (such as the hyperplane classifier), while being mathematically compelling because of their ease of analysis, have limited computational power and thus limited real-world value [9]. In general, complex real-world applications require a learning machine with much more expressive power.

One proposed solution to this problem was to create a network of simple linear classifiers (in the form of neural networks) and thus be able to represent nonlinear decision surfaces. However, neural networks have a number of inherent problems, including local minima and many tunable parameters. In addition, it is very complex to analyse a neural network mathematically.

Another solution is to map the input vectors into a richer (usually high-dimensional) feature space where they are linearly separable using a nonlinear mapping \( \phi \). In feature space, build a separating hyperplane using a well-understood linear learning machine such as the optimal hyperplane classifier (see Figure 3). This yields a nonlinear decision surface in input space and is the approach taken by Support Vector Machines.

As we have already noted in Section 2.1, the optimal hyperplane classifier uses only dot products between vectors in input space. In feature space this will translate to \( \langle \phi(x) \cdot \phi(y) \rangle \). Clearly, this is
very computationally expensive, especially if the mapping is to a high-dimensional space. Boser, Guyon and Vapnik [2], showed that a rather old trick [1]—kernel functions—can be used to accomplish the same result in a very simple and efficient way. A kernel is a function $k(x, y)$ that given two vectors in input space, returns the dot product of their images in feature space

$$k(x, y) = \langle \phi(x) \cdot \phi(y) \rangle. \quad (5)$$

There are several different kernels, choosing one depends on the task at hand. One of the simplest is the polynomial kernel $k(x, y) = \langle x \cdot y \rangle^d$. For example, taking $d = 2$ and $x, y \in \mathbb{R}^2$

$$\langle x \cdot y \rangle^2 = (x_1y_1 + x_2y_2)^2$$
$$= (x_1y_1 + x_2y_2)(x_1y_1 + x_2y_2)$$
$$= (x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2)$$
$$= (x_1^2, x_2^2, \sqrt{2}x_1x_2)(y_1^2, y_2^2, \sqrt{2}y_1y_2)$$
$$= \langle \phi_1(x) \cdot \phi_1(y) \rangle$$

$$\quad (6)$$

defining $\phi_1(x) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$.

4 Support Vector Machines

Support Vector Machines are nothing more (or less) than linear learning machines expressed in dual form that map their input vectors to a feature space by the use of kernels and compute the optimal hyperplane there.

If we take Equation 4, which is the decision function for the optimal hyperplane classifier in dual form and apply the mapping $\phi$ to each vector it uses, we will get

$$f(x) = \text{sign} \left( \sum_{i=1}^{\ell} y_i \alpha_i \langle \phi(x) \cdot \phi(x_i) \rangle + b \right). \quad (7)$$
As already mention in Section 3 the explicit mapping to feature space is not desirable, since it is very computational expensive. We will therefore use kernels which will give us a nonlinear decision function of the form

\[ f(x) = \text{sign} \left( \sum_{i=1}^{\ell} y_i \alpha_i k(x, x_i) + b \right). \]  

(8)

The SVM algorithm is thus based on statistical learning theory, while being practical since it reduces to an optimisation problem with a unique solution. Up to now we have only considered the case of classification (pattern recognition). A generalisation to regression, that is, having \( y \in \mathbb{R} \), can be given. In this case, the algorithm tries to construct a linear function in the feature space such that the training points lie within a distance of \( \varepsilon > 0 \). Similar to the pattern-recognition case, this can be written as a quadratic programming problem in terms of kernels [13] (see [6] for details).

5 Final Remark

Support Vector Machines have been applied to many real-world problems, producing state-of-the-art results. These include text categorisation [7, 8], image classification [5, 10–12], biosequence analysis and biological data mining [3] and handwritten character recognition [2].

References


Abstract. This effort sets out to outline a research domain of academic and commercial relevance as well as the establishment of a possible research trend in the field of software engineering. The OO approach has established itself as a widespread and effective paradigm for modern software development. Many aspects of OO development are methodologically supported and procedural and representation standards are clearly defined. Certain activities within OO development remain suited for both automated and manual interpretations. It is also a fact that many system descriptions start off as natural language accounts of business processes, rather than semi-formalised data-flow or use-case models. It is therefore being proposed that a direct-from-text reliable and complete conversion method with governing standards can be defined to automate as necessary the class derivation activity, therefore decreasing the overall development effort and error-introduction probability without effecting objectivity within the OO development process. Such a conversion method would also allow more accurate rapid prototype generation at the earliest development stages. In theory, this would enable developers to automatically generate better quality “first-cut” GUI prototypes directly from textual system descriptions.

1 OO Development Effort

The effectiveness of engineering modern software systems has been an evolving multifaceted issue for a relatively long period of time [6] [7]. Several development paradigms exist each highlighting a “pathway” through the software engineering process [15]. An effective paradigm in modern software development is the Object-Oriented (OO) paradigm [4] [21]. The OO approach to system development is young enough to elicit research interest and yet, trodden enough to have proven its practical (commercial) aptness [11]. The advantages of OO-developed systems are discussed in numerous publications and are in particular elegantly outlined in Therekhov’s paper [24]. Therekhov also sensibly justifies the effort for converting non-OO-developed (“legacy”) systems into OO ones. The marked implications that the OO development paradigm has on software reusability and component-based software development, further highlight its potential.

2 OO Development Issues

One of the more essential requirements for effective OO system development is the adoption of a common development platform [20]. Such a platform should include models, procedures and guidelines. Without a common development platform, much effort would be misdirected causing many of the benefits which can be gleaned through the OO paradigm to be lost. In the course of the author’s work experience within private companies supporting software development, it has been
repeatedly noted, that individual programmers exhibited definite but naturally specialised (hence limited) effectiveness. This situation is adequate for development of systems of pre-determined sophistication or the specific customisation of existing highly-sophisticated systems. However, once projects of considerable calibre are undertaken from inception, basic effort communication becomes crucial [16] [23] and the lack of a common development environment and standard modelling procedure considerably reduce development efficiency. This is an emerging phenomenon encountered by the author in many large and well-established companies in Malta.

3 OO Development Support with UML

One of the most prevalent development environments supporting the OO paradigm is the Unified Modelling Language (UML) [9] [18]. UML relies on a powerful combination of graphic and textual syntax to model various aspects of a system from an OO perspective. UML has given rise to several development processes closely based on it and is fast becoming an industry-standard for OO-based system development – mainly, but not solely, due to the continuous “fine-tuning” and enriching input of a host of prestigious commercial partner establishments as well as its ever-increasing user-base. One such development process currently enjoying widespread popularity is the Unified Software Development Process (USDP) or the Rational Unified Process (RUP), or simply the Unified Process (UP) [11]. USDP is made up of the following development phases:

- Inception
- Elaboration
- Construction
- Transition

The phase where most of the design effort is concentrated is the Elaboration phase. This phase is made up of the following OO analysis steps:

- Class modelling
- Dynamic modelling
- Functional modelling

UML coverage of the Software Development Life Cycle (SDLC) ranges from the most abstract forms of system representation up to and including physical system models [3]. The research introduced through this document is mainly directed at issues in the Class modelling step.

4 Class Identification Issues

Although UML supports all the main phases in OO software development through precisely defined frameworks, procedures and guidelines, it nevertheless lacks any form of formal enforcement. This is particularly relevant to the traditionally more subjective activities in OO development. A typical example of such an activity would be the derivation of classes from a natural language description of a system [2]. This is the core activity of the class modelling step. Through the author’s personal observation, it has been noted, that considerable system development errors are the result of incorrect or inappropriate derivation of classes from textual system descriptions. In the case of inexperienced developers or students, this form of error is manifested as missed, inconsequential
or wrongly derived classes. In the case of more experienced developers, the same form of error can be manifested as iterated (“multi-pass”) class analysis leading to substantial development effort dissipation.

The most common approach adopted to try and limit the subjectivity of class identification is to offer developers a modelling tool that would allow system functionality to be specified at a high level of abstraction thus leading the developer to identify classes in full or partial fulfilment of the modelled functions. Such a modelling tool is the classical Data Flow Diagram (DFD) at context and first levels and later, the UML Use-Case Diagram (UCD) [19] [17]. However, DFDs at context level tend to be too abstract to offer any useful information regarding class identification, and DFDs taken to level one would have already required considerable effort and decision making in the first place. On the other hand, UCDs are often misinterpreted or trivialised based on the assumption that every use-case is realised by a class. In other words, the relationship between UML UCDs and UML Class Diagrams is not always straightforward and can be the source of development errors [25].

5 Prototype Generation Issues

Prototypes are one of the most helpful tools for both developers and users alike [10] [26]. Several valid approaches to generating prototypes exist [22] [27]. Most can be commonly summarised as follows:

1. Scope the system (with user input);
2. Determine the scenarios (with user input);
3. Model scenarios through UCDs;
4. Confirm scenarios (with user input);
5. Reconcile user viewpoints;
6. Produce prototypes.

Some fundamental issues can arise with the above generic (traditional) approach, namely:

– The user is really independently in the picture only while system description is in terms of natural language. Once descriptions move into the UCD domain, users will generally rely on developer interpretation of technical models to understand proposed system behaviour and then provide any corrective input. Users feel intimidated by this.

– Considerable effort is already in place before any form of tangible prototype is produced. This generally leads to predominance in the creation of “worth-it” prototypes.

– Prototypes created with this traditional approach tend to be biased towards whatever it is that the developer wishes (consciously or sub-consciously) to show.

6 Proposed Framework

The approach being proposed in this work is one that will allow users to feel comfortable in their system explanation and understanding and developers to feel comfortable and secure in their derivations.

The proposed technique is one that extends the basic traditional noun-verb analysis technique for class/method identification as pioneered by Russell J. Abbott [5]. The noun-verb class derivation
technique is only really effective when the relationship of classes to nouns is (or is close to) one-to-one [14]. The chosen grammar-based analysis of text is based on the breakdown presented by Dennis [8], which in the author’s experience is one of the most comprehensive and credible. Dennis presents the following grammatical analysis mapping:

- A common or improper noun implies a class
- A proper noun or direct reference implies an object (instance of a class)
- A collective noun implies a class made up of groups of objects from another class
- An adjective implies an attribute
- A “doing” verb implies an operation
- A “being” verb implies a classification relationship between an object and its class
- A “having” verb implies an aggregation or association relationship
- A transitive verb implies an operation
- An intransitive verb implies an exception
- A predicate or descriptive verb phrase implies an operation
- An adverb implies an attribute of a relationship or an operation

It is well known that numerous lexical parsing techniques exist each having their own specialised application. In the case of extracting all or the key parts of speech from the above list, basic lexical parsing is all that is envisaged. The well established technique of parsing by chunks [1] is a possible adequate method. However, the choice, justifications and testing involved in the actual choice is a matter for future consideration and can indeed constitute a research topic in its own right.

One of the most comprehensive and well structured procedures for class/object and interface identification is presented by Lee [14]. It is being proposed, that given Dennis’ grammar-based breakdown, Lee’s identification procedures and some additionally derived rules, it would be possible to automatically generate direct (i.e. one-to-one noun-class assumption) class lists, proceed to optimise the direct class list, and even generate a first-draft class diagram, which is, in effect, the optimised classes with defined inter-relationships. It is also being proposed, that consideration be given to the possibility of generating first-draft GUI prototypes from class diagrams (attributes, operations and relationships). It can be argued, that having a fully specified class diagram (even a first attempt), enough data is present to infer upon the structure and behaviour of the corresponding GUI.

Figure 1 shows a generic overview of the class and prototype derivation technique being proposed.

![Fig. 1. Generic overview of class and prototype derivation technique](image-url)
To conclude The effort outlined in this document can be classified as applicative and innovative. Applicative in the sense, that existing techniques are integrated and applied in a focused manner towards the betterment of the overall software development process. Innovative in the sense, that new interpretation and derivation methods can be properly researched and developed, which can eventually lead to levels of automation and objectivity so far unachieved.

The effort proposed is comprised of more than one possible research thread. One can clearly distinguish research threads in lexical analysis, data and behaviour derivation and correlation, interface inference, and possibly, notation (and semantic) development.

References

Automatic speech recognition is the technology that allows humans to communicate to a machine through spoken utterances instead of the more conventional means such as keyboards. The spoken signal is captured and analysed in a series of one or more steps, and action is taken depending on the final representation. Speech recognition inherits much from the cognitive sciences since most approaches try to first understand how humans perceive and interpret speech, and then build a model that best represents human perception.

While models for high performance speech recognition have been studied extensively for many decades, it is only recently that we have seen some impressive results. The main reason for this is that unfeasible models for high performance systems designed in the past have become feasible due to the increase in the available processing power. Modern speech recognisers extract features from the time-varying signal that discriminate between the different units of speech, being affected only slightly by the pitch or manner of the speaker. The features are then passed through one or more trained Hidden Markov Models (HMMs) for the classification of a phonetic sequence, and ultimately the utterance itself [18, 13].

It should be clear that a speech recognition system is highly dependent on the base language. The reasons for this are several, but mainly include (i) the different subset of phones that optimally model a language (ii) the approximately constant language statistics representative of the language, extracted from an annotated speech corpus. All high performance speech recognisers are based on probabilistic models (e.g. HMMs), and thus the importance of (ii) must not be underestimated.

While extensive linguistic study has been performed on the Maltese language in the classical sense, including grammatical studies such as [7], historical studies [3, 8, 10], dialectical studies [2, 9, 1] and phonological studies [6, 4], there currently lacks material for computational resources in Maltese. At the extreme we do not have an organised and central corpus for the Maltese language. Projects requiring such material usually have to consider the task of collecting the different fragments from various sources. With regards to resources related to speech, such as annotated speech corpora, the situation is practically non-existent. Projects in this field [15, 12] have usually relied on developing their own small resource “islands”, limited due to the nature of the laborious task involved in manual annotation. Currently literature includes the development of a speech synthesiser for the Maltese language [15], a tool for automatic annotation of speech samples [12], the development of a computation method for the automatic generation and labelling of a Maltese computational lexicon [11], and projects such as a study in the application of a deterministic spell checker for Maltese [14] and the application of a stochastic spell checker for Maltese [17], and finally an automatic machine translation system from English to Maltese focusing on weather announcements [4]. Various papers have also been published to introduce the ideas, concepts and goals of the Maltilex project, a joint project by the Department of Computer Science & A.I. (Faculty of Science), the Department of Communications and Computer Engineering (Faculty of Engineering) and the Institute of Linguistics, whose major task is the development of a POS-labelled Maltese computational lexicon [19, 20, 16].

In general, the aim of this work is to continue developing on the existing resources, in particular on those concerning speech. A phonetic analysis of the language will be performed from a computa-
tional point of view with the aim of developing and evaluating the first simple speech recogniser for the language. However the project will not stop at the academic level. The intention is to deploy the speech recognition modules on top of an Interactive Voice Response system (IVR), as used extensively by telephony applications (e.g. voice mail). Studies have shown [21] that using a speech interface to IVRs increases successfully serviced customers, reduces customer mistakes, improves routing accuracy, and reduces the routing time. All of these factors will decrease the running costs of an IVR system, in particular those applied to call centres, [22] and provide a much more interesting experience to customers [21]. This naturally yields to a return and eventually profit on the initial investment cost.

From a technical point of view, the GSM mobile channel will be given special consideration due to the fact that GSM encoding is a lossy compression standard that introduced noise to the original waveform [5]. The final deliverable will be a toolkit by which speech-driven IVRs can be easily developed on top of the classical touch-tone system. As a case study a practical IVR implementation will be developed that will focus on the recognition of single words and sequences of continuously spoken digits such as telephone numbers. The actual particular application has yet to be determined since one has to focus the available resources into a commercially viable direction.

The evaluation of the system is expected to provide answers to questions based on [21] such as whether the Maltese public prefers to talk to speech recognition systems in general rather than through touch-tone menus, whether the Maltese public finds the experience of using this particular system better or not than an older touch-tone driven version depending on the application, rate the system on a standard 5-scale Likert-scale and questions of an original nature, such as whether the Maltese public prefers using Maltese or English as a communication medium to IVR systems. An analytical study of the recordings will be performed to determine the extent to which Maltese phonemes are used by Maltese nationals when speaking English words.

Minor tasks include improving the current version of the text-to-phoneme transcriber to extract and publish statistics on Maltese phonemes, both when considered on their own and also in the context of their surrounding phonemes.

References


A Summary of Research in System Software and Concurrency at the University of Malta: I/O and Communication

Joseph Cordina
Department of Computer Science and AI,
University of Malta

Abstract. Traditional operating systems and commodity hardware are never used to their full potential due to underlying design limitations. Applications that make use of blocking system calls incur large overheads on the operating systems and in turn end up wasting CPU resources. In addition, traditional solutions are not adequate for high-performance networking. In this report, we present a summary of the research conducted by the System Software Research Group (SSRG) at the University of Malta. We discuss some of the solutions we have developed and pinpoint their effectiveness to solve each of the above problems.

1 Introduction

Facilities commonly offered by operating systems, such as memory protection and I/O resource usage, often result in huge bottlenecks to applications demanding huge computational resources. The traditional way of accessing resources is through system calls. Each system call demands switching of protection boundaries which is quite computationally expensive. In addition, CPU time and consequently application time is wasted when waiting for these resources to terminate their requests. Such slow resources are disk usage, network communication and even memory access.

The System Software Research Group[15] (SSRG) was set up in the Department of Computer Science at around 1999. Its main aim is to conduct research in the area of System Software. The above problems proved to be an ideal direction for further research. In [22] Vella proposes several methods that the SSRG has worked on to improve performance at the application level. Some of these optimisations are based on user-level multithreading techniques. Unfortunately user-level thread schedulers fail in their utility when invoking system calls, in that the user-level thread scheduler becomes blocked until the underlying kernel thread itself is re-scheduled by the kernel. The propose several solutions to this problem developed within the research group and analyse their performance.

Computing today is tightly dependent on network performance. In tests performed in [6] we have shown the limitations proposed by current commodity hardware in dealing with high bandwidth networking. Thus this research group has also concentrated on removing the bandwidth bottlenecks while still using commodity hardware. We show that cheap hardware solutions are able to cater for highly demanding network applications.

2 Operating System Integration

Any application that makes use of system resources such as disk I/O and networking needs to make calls into the underlying kernel. The reason for this barrier is due to current operating systems that
protect the above application from access intricacies and to protect one application from another. Unfortunately when accessing slow resources, the application remains blocked until the resource can service the request. In the mean time, other kernel processes or threads can continue executing.

In highly demanding applications that make use of such resources, the common alternative is to make use of multiple kernel entities that will be utilised whenever the running thread gets blocked. This provides better CPU usage. Yet this is not ideal due to the expense of creating new kernel threads and the expense of switching to other kernel threads. In addition, when utilising user-level thread schedulers, when one user level thread makes a blocking call, all the other user level threads are not able to continue executing. Within the SSRG we have investigated various solutions to this problem. One of the most basic solutions is the use of wrappers. This is a modification of potentially blocking system calls, such that such calls invoke a second kernel thread that continues to execute other user-level threads that are not blocked. Whenever the blocked user-level thread unblocks, the second kernel thread halts and the original user-level thread continues executing. Wrappers provide an adequate solution when the system call actually blocks. Unfortunately system calls that do not block (for example when a read call is issued that already has data locally in a memory buffer) cause unnecessary overheads. In addition, wrappers provide a solution which is not transparent to the application programmer.

In addition to system calls, the application may also block whenever accessing memory pages that are not readily available (memory pages have to be initialised or have been swapped out to disk space). Such page faults still cause the application to block for an undefined amount of time, wasting CPU time. Wrappers are obviously not capable of solving such problems. Within the SSRG we have concentrated on providing solutions that achieve high performance while being transparent to the programmer.

### 2.1 Activations

We have developed a solution that is capable of solving the above problems. Through Linux kernel modifications, we were able to use a mechanism such that whenever a system call blocks, a call is made from the underlying kernel to the blocked application. This kind of information allows the application to react accordingly to the changing environment of the operating system. By making use of thread pools within the kernel and utilising user-level thread schedulers, we were able to solve most of the problems associated with blocking calls. Our results\cite{3} have shown that this solution is able to achieve high performance when applied to an application that is dependent on a large number of potentially blocking system calls such as a web server. This solution was also extended to be applied to SMP architectures. Unfortunately due to limitations in the underlying kernel modifications, this solution cannot be applied to blocking due to page faults.

### 2.2 Shared Memory Asynchronous Communication

In comparison to the above solution, we have developed a further mechanism whereby a shared memory region is used by the application and the underlying kernel to pass information to each other. This mechanism offers a solution scalable primitive that was applied to the above blocking call problem and also to the extended spinning problem\cite{18}. By making use of this asynchronous mechanism, the kernel can create new kernel threads whenever a blocking system call occurs and can also inform the user-level thread scheduler whenever a system call unblocks. This solution was found to be comparable in performance to the activation mechanism while providing additional scalability.

\footnote{This is a problem that occurs whenever several processes make use of a shared lock to execute critical sections in a multiprogrammed environment\cite{19}}
3 Network Communication

Most applications today depend heavily on network connectivity and fast bandwidth. While CPU speed is ever increasing, advances in network technologies, interconnecting architectures and operating system software has not kept pace. The SSRG has performed several tests to analyse bottlenecks within the connection pipeline. We have found that that on commodity platforms the main bottlenecks are the operating system network stack, common interconnecting architectures such as the PCI and the network protocol itself. We set out to prove that gigabit connectivity can be achieved on commodity platforms and Ethernet standards.

3.1 Ethernet Bandwidth

Various tests were carried out by Dobinson et al.\(^2\)\(\cite{13}\) to analyse network traffic on Ethernet hardware. The test controller used was the Alteon Tigon controller\(\cite{16}\), a gigabit Ethernet controller that allows custom software to be uploaded and executed on board the card. It was found out that making use of larger Ethernet packets\(^3\) it was possible to achieve gigabit data bandwidth on the physical line.

The SSRG then concentrated on solving the problem in other layers of the communication pipeline. In his project, Wadge\(\cite{24}\) made use of the processing power of the Alteon card by modifying the firmware such that bottlenecks on the PCI were bypassed for Ethernet packets. This was performed by transferring large chunks of data through the PCI instead of the traditional 1.5K packets. When using Ethernet packets, we achieved almost gigabit bandwidth, a first when making use of 33MHz, 32 bit PCI. In addition, the group took ideas from MESH\(\cite{2}\), a user-level thread scheduler integrated with Ethernet communication, to make use of an area of memory accessible only to the application thus allowing user-level access to the Alteon’s frame buffer.

3.2 TCP/IP bandwidth

The group still desired to extend the above results further to bypass restrictions found in most networking stacks of traditional operating systems. It was found that the cost of crossing the protection barrier between the application and the underlying kernel, coupled with several data copies that are made within the kernel for TCP/IP connection, severely degrades the point to point bandwidth. In addition, we found that CPU utilisation is substantial for high bandwidth networking to the point that on commodity hardware, applications are devoid of CPU resources. Thus we have investigated the development of user-level networking, where the kernel does not take part in the communication process\(\cite{6}\). This was achieved through the use of the Alteon Ethernet Controller and an area of memory reserved to the application. While this mechanism does not offer the traditional protection facilities provided by the operating system, we felt justified in bypassing this restriction due to the high-demand environment of this application. We have also developed a TCP/IP stack, solely at the user level and using zero-copy communication. We managed to achieve very high bandwidth rates with very low CPU utilisation on relatively slow commodity hardware, showing that demanding applications can be developed and executed on inexpensive hardware.

\(^2\) In collaboration with CERN and the SSRG
\(^3\) Known as Jumbo packets that are 9K large instead of the standard 1.5K packets
3.3 User-Level Threads and Networking

To make use of the high-performance techniques proposed in [22] and above, the research group has developed certain extensions to SMASH[8] (a user-level thread scheduler). SCOMM[4] is a product that allows CSP-like channel connectivity on the network. It makes use of TCP/IP yet was developed to allow integration of other network drivers underneath. It solves the blocking problem of traditional communication calls by allocating the responsibility of network communication to another kernel thread, thus avoiding the user-level thread scheduler from blocking. In addition Nickovic[17] has further enhanced SCOMM by designing and implementing shared variables though this communication protocol. This mechanism was built without requiring any central servers for variable consistency.

4 Conclusion

We have shown the various areas that have tackled by the System Software Research Group in terms of I/O and communication. We solved the problem of applications that make use of blocking system calls and applied our solutions to maximise CPU usage. In addition, we have developed several high-performance networking solutions that cater for high bandwidth and low CPU usage. While various projects have been developed from this research, and most of the work is at par or better with the best research done in this area, there is still more work that we plan to achieve. We plan to integrate our user-level TCP/IP package with SMASH. When integrated, this would provide a high-performance solution towards computation with high-bandwidth networking at very little cost. In addition we plan to work towards building higher level applications on top of SMASH. We aim to be able to apply peer to peer algorithms[14] and distributed data structures[7] to provide a framework for high-performance distributed computation and space storage.

The System Software Research, while burdened by limited resources, both in terms of hardware and man power, has managed to achieve products of very high quality. We have developed several novel solutions that are capable of servicing a large number of demanding request making full use of the capabilities of the underlying hardware.

References

1. S. Abela. Improving fine-grained multithreading performance through object-affinity scheduling. B.Sc. I.T. Final Year Project, Department of Computer Science and Artificial Intelligence, University of Malta, June 2002.
4. S. Busuttil. Integrating fast network communication with a user-level thread scheduler. B.Sc. I.T. Final Year Project, Department of Computer Science and Artificial Intelligence, University of Malta, June 2002.


17. D. Nickovic. Distributed shared variables for user-level fine grained multithreading. B.Sc. I.T. Final Year Project, Department of Computer Science and Artificial Intelligence, University of Malta, June 2003.

18. I. Sant. An asynchronous interaction mechanism between the kernel and the user application. B.Sc. I.T. Final Year Project, Department of Computer Science and Artificial Intelligence, University of Malta, June 2003.


Text-To-Speech Technologies for Mobile Telephony Services

Paulseph-John Farrugia

Department of Computer Science and AI, University of Malta

Abstract. Text-To-Speech (TTS) systems aim to transform arbitrary\textsuperscript{1} textual input into spoken output. At first glance, this may seem a relatively simple task of determining the phonetic sounds of the input and outputting a corresponding sequence of audible signals. However, it is in fact quite a difficult task to produce intelligible and natural results in the general case. This is due to linguistic and vocalization subtleties at various levels that human speakers take for granted when interpreting written text. In fact, the task requires a considerable grasp of both Natural Language Processing and Digital Signal Processing techniques.

The potential application of such functionality is varied, including its use for language education, as an aid to handicapped persons, for implementing talking books and toys, for vocal monitoring and other man-machine communication facilities. The area is currently being explored in order to address its application for the Maltese language\textsuperscript{2} within the context of the mobile communications industry.

This paper’s main aim is to provide a brief overview of current TTS approaches and techniques, and the way these may be implemented. For further insight, reference should be made to the selected bibliography.

1 Introduction

In general, the transduction process from text to speech is carried out through a sequence of readily recognizable steps that provide an increasingly detailed (narrow) transcription of the text, from which the corresponding spoken utterance is ultimately derived. These steps can be divided into two blocks—the Natural Language Processing (NLP) block and the Digital Signal Processing block. This organization is shown in Fig. 1 (Adapted from [3].)

These steps should not be thought of as ‘filters,’ but as processes that incrementally augment the information derived from the input and place it on a commonly accessible Multi-Level Data Structure (MLDS). The ultimate aim is to derive sufficient information on the MLDS so as to be able to derive an intelligible (i.e. readily understandable to a listener) and natural (i.e. comparable to a human speaker as regards intonation and prosody.)

2 The NLP Block

The NLP block consists of those processes that are responsible for analyzing the text in order to extract a sufficiently detailed narrow phonetic transcription that can eventually be used for the

\textsuperscript{1} As opposed to ‘canned’, such as is the case with pre-recorded, concatenated IVR messages.

\textsuperscript{2} Although attempts are made to unify the field across languages, TTS implementations need to cater for language specific phenomena.
Fig. 1. TTS Processes
DSP, or synthesis, block. This may be considered the more language-specific block, as it needs to derive a concrete description of how the textual input should sound (without actually deriving the corresponding sound per se.)

The processes will now be considered in order.

2.1 Pre-Processor

The pre-processing module is responsible for transforming the text into processable input, that is, into a list of words. Among other things, it is responsible for:

- recognizing and spelling out acronyms, which is not so straightforward when one considers different interpretations of acronyms such as, for example: CNN, SARS, CSAW
- spelling out numbers, taking context into consideration, as in, for example: 25, Lm 25.00, 15 klteb.
- resolving punctuation ambiguity as, for example between a full stop identifying the end of a sentence and one at the end of an abbreviation.

2.2 Morphological Analyzer

The morphological analyzer utilizes lexical information in order to derive a morphological parse for each word, and consequently identifies its possible parts of speech.

2.3 Contextual Analyzer

The contextual analyzer considers the surrounding context of the words in order to limit the possible parts of speech to a minimal, likely set.

2.4 Syntactic Prosodic Parser

The syntactic prosodic parser organizes the lexically analyzed word list into identifiable boundaries (sentences, clauses and phrases.) This text structure will then be utilized in order to derive an appropriate prosodic representation.

2.5 Letter to Sound Module

The Letter to Sound (LTS) module, (also referred to as the grapheme to phoneme module) derives a phonetic transcription for the input. Once again, upon first inspection this may seem like a trivial process of looking up entries in a pronunciation dictionary. However, this is in fact not feasible, and the process is a complex one for the following reasons:

- Due to morphology, no pronunciation dictionary could be considered complete in respect of providing an entry for every possible word form. An ordinary dictionary, for instance, usually only provides a pronunciation for a basic word form. Thus, even if a pronunciation dictionary is used, morphophonological rules will be required in order to account for derived words.
Some words will lend themselves to more than one pronunciation depending upon their part of speech or context. These are referred to as heterophonic homographs, examples of which are sur (/sur/ or /suːr/) and bajjad (/bejˈjaːd/ or /bejˈjaːd/).

The pronunciation of a word is affected by its surrounding textual context, in particular at the word boundaries. Hence, it is may not be possible to store an absolute pronunciation for a word independent of context.

Even the most complete pronunciation dictionary would not contain entries for new words which make their way within a language, or for other classes of words such as proper names.

Two main approaches to this problem can be found. The first, dictionary-based approaches, do indeed utilize a pronunciation dictionary in order to provide as much language coverage as possible. Entries are usually stored in the form of morphemes, with pronunciation of derived forms obtained using morphophonemic rules. Unknown words are transcribed by rule. The rule-based approaches invert this by transcribing most of the input by general grapheme to phoneme rules, and only keep a relatively small pronunciation dictionary for known exceptions.

2.6 Prosody Generator

The term prosody refers to the speech signal properties such as intonation, stress and rhythm. The effects of prosody have a direct influence on how the message represented by the text is conveyed, differentiating, for instance, between a statement or a question (by means of changes in intonation) or providing focus on the subject matter (by means of appropriately placed stress.)

The prosody generator, then, is responsible for identifying the intonational phrases corresponding to the text and assigning the appropriate prosody contour.

3 The DSP Block

The DSP block, depicted in an over-simplified manner in Fig. 1, is, at an intuitive level, the programmatic counterpart of the human speech reproduction organs. It utilizes the narrow phonetic transcription derived from the previous block in order to generate the actual speech waveform that can be audibly reproduced.

Two main approaches are encountered for this task. The first is the rule based approach, which attempts to provides synthesis through the modelling of the human speech reproduction process as the dynamic evolution of a set of parameters. This approach has a number of benefits. For instance, it is speaker independent, that is, the voice is completely synthetically generated, and can hence be altered at will by modifying the appropriate parameters. However, this approach is complex and takes a long development effort due to the difficulty in deriving the appropriate set of rules.

The second approach is the concatenative one, which utilizes a speech database consisting of pre-recorded elementary speech elements (typically diphones, two sequential phoneme units) as the basis for synthesis. In practice, the output is generated by concatenating a sequence of elementary speech elements corresponding to the phoneme sequence identified by the NLP block. DSP algorithms are then applied in order to smooth the resulting waveform and apply the intended prosody. By the contrast to the rule-based approach, concatenative synthesis is strictly speaker dependent, as the speech database is generated from recordings of appropriately chosen text by a professional speaker.
4 Research and Application

Research is being carried out in order to apply TTS techniques in order to be able to develop a system for the Maltese language, with a practical application within the context of mobile telephony in providing extended services such as SMS to voice mail. Particular emphasis is intended on the prosody generation phase. The aim is to utilize previous work on TTS for Maltese ([7]) and Maltese prosody ([10]) in order to develop a framework for more natural synthesis results.

References

Open-domain Surface-Based Question Answering System

Aaron Galea
Department of Computer Science and AI, University of Malta

Abstract. This paper considers a surface-based question answering system for an open-domain solution. It analyzes the current progress that has been done in this area so far, while as well describes a methodology of answering questions by using information retrieved from very large collection of text. The solution proposed is based on indexing techniques and surface-based natural language processing that identify paragraphs from which an answer can be extracted. Although this approach would not solve all the problems associated with this task the objective is to provide a solution that is feasible, achieves reasonable accuracy and can return an answer in an acceptable time limit. Various techniques are discussed including question analysis, question reformulation, term extraction, answer extraction and other methods for answer pinpointing. Besides this further research in question answering is identified, especially in the area of handling answers that require reasoning.

1 Introduction

Information Retrieval (IR) techniques were developed with the intention of helping users to find the information required. The search engines developed for this purpose were adequate for some time but they are now finding their limitations. Usually search engines return a huge list of relevant documents for a user query, which still requires the user to skim through the documents to find the necessary information. The invention of Question answering (QA) systems is to avoid this user overhead and present them with the direct answer to the question. For example if a user asks the question “Who is the president of Malta?” as an answer he/she gets the name of the Maltese president rather than a whole range of documents describing everything that might be related to the question. In this paper an attempt is made to achieve a QA system that can answer users’ questions with relatively good accuracy and provide the ability to answer factual questions supplied by the users.

This paper is structured as follows. In section 2 the previous work done in this area is discussed. After this an overview of the approach followed for our system is presented in section 3. Section 4 will then describe some modules making up the system. Section 5 presents the evaluation results of the system compared with others. Finally Section 6 concludes the paper and gives future enhancements.

2 Approaches

Research in question answering systems is not something innovative since interest in them started from the era of knowledge bases. However they were always employed in closed domains [1, 2]. The interests have now shifted into scaling these QA systems to an open-domain solution and enhance search engines to provide this granularity of detail to users. Achieving this is still not widely available even though some attempts have been made like Google QA\footnote{https://answers.google.com/answers/main} and Ask Jeeves\footnote{http://www.ask.com}.
Open-domain question answering is being attempted either by using deeper-level or surface-based approaches. Deeper-level approaches offer the ability to handle more complicated questions but at the same time has limitations since a knowledge base has to be constructed. Knowledge in this kind of systems is represented either in KIF or CGIF format [3] (pg 427). One problem with this kind of QA systems is that an ontology has to be built so that it can reason about the world being modeled [4, 3]. An attempt to build an ontology for these situations is being handled by Cyc3 while another research makes use of a linguistic resource to include background knowledge [5]. Despite these efforts such an approach is still unfeasible to implement for open-domain solutions. This is the same idea of the semantic web4 but in the case of QA systems it is impossible to create an ontology for every document used.

The other approach that offers more chances of success is a surface-based approach. One problem with such an approach is that they can handle only factual question that appear directly in text. When asking questions that require reasoning a deeper-level approach is the only alternative. In fact most of the QA systems competing in TREC applied this approach [6–9] and offers a number of advantages. First of all performance can be measured by using the TREC evaluation method [10]. These systems as well are based on current state-of-the-art technology like parsers [11–13], entity name recognizers [14–16] and search engines [17, 18]. However such QA systems can be extended to handle complex questions. In LASSO [4], a surface-based QA system was extended to incorporate a theorem-prover to verify whether a piece of text is an answer to a question.

Considering the benefits of a surface-based approach to question answering and the extensionality it provides, in this paper a QA system for educational purposes is presented based on this technique. The motivation behind this work is to build a QA system to help students with their difficulties in particular subjects. Apart from this it also shows how an FAQ module that detects semantically equivalent questions can improve system accuracy. We employ the Winnow algorithm [19] for question classification rather than the traditional rule-based approach followed in other QA systems [6–9]. One major difference in this QA system is that it assumes that a question could have been previously met in a variant form while other QA systems assumed that all previously submitted questions were never seen.

3 Overview Of Approach

Figure 1 represents the approach followed in our QA system. Initially a user submits a question, which gets processed by the FAQ module. It tries to identify semantically equivalent questions that have been asked before. If it succeeds to find a semantically equivalent question its answer is returned. On the other hand if it fails, the questions is forwarded to the Question identification module whose purpose is to classify the question. For example, if the question asked is “Who is Vidocq?” then the question should be classified as a Person. This is useful during later stages when entity extraction rules are run on the text segments retrieved. The user question is further processed by a Term Extraction module. Its objective is to extract useful terms to search in the indexed documents. The Terms extracted together with the classified question are passed to an Answer Extraction module. Its purpose is to identify a text segment that might contain an answer to a question. It uses the terms to identify a number of documents that might contain the answer. These documents are further processed to extract sentences that can form potential answers. Finally entity extraction rules are run on the text segments retrieved to better score the sentences that can form an answer. Since the system supports an FAQ module that is constantly being updated the system also supports feedback from the user. This allows a list of questions to be accumulated by

3 http://www.cyc.com/products2.html
4 http://www.w3.org/2001/sw/
the FAQ module so that future users asking similar questions would get the answer quicker rather than taking the time consuming task of extracting answers from documents.

In the next section these modules are described in more detail.

![Figure 1: Answer Extraction process](image)

### 4 Architecture

The system architecture presented in figure 2 represents the modules mentioned in the previous section together with other tools being used by the system. These modules are described in the following subsections.

![Figure 2: System Architecture](image)
4.1 FAQ (Frequently-Asked question) modules.

The FAQ module purpose is to identify semantically equivalent questions. The algorithm applied is borrowed from [20]. Its attractiveness is that it does not depend on well-formed questions. This is something important for our QA system since we cannot assume that the questions supplied are grammatically correct. This FAQ module tries to improve the system performance without needing to use expensive operations like answer extraction. If the FAQ module fails the rest of the modules come into play to extract an answer from the indexed documents. The first module to start the whole extraction process is the Question Processor (QE) module.

4.2 QE (Question Processor) module

The QE processor purpose is to classify the question submitted and as well extract key terms from the question. Question classification employs a trained Winnow algorithm using the SNoW toolkit [21]. The term extraction part uses the question to extract terms that are either quoted terms, noun terms and does not belong to a stop word list. Besides this feature it also expands certain terms based on word synonyms. Once this information has been extracted it is passed to the Answer Extraction (AE) module.

4.3 AE (Answer Extraction Module)

The Answer Extraction module uses the terms extracted to search for documents that contain the terms. It uses an approach similar to the Lasso question answering system [22]. In order to reduce the documents returned a Paragraph N operation is applied that checks whether the terms appear in N consecutive paragraphs. Once this has been completed, paragraph filtering selects segments from the paragraphs that may contain the answer. The final stage is to score the segments retrieved based on the key terms present, the order of the key terms and entities present in the text.

5 Evaluation

We evaluated the question answering system in two stages. For the FAQ module we used the TREC-9 data that is available. However evaluating the AE module was performed on our own collected data since the TREC evaluation data was not available. This data consisted of comprehensions text together with their questions.

A total of 124 questions were used for evaluating the FAQ module purposes. Out of this 124 questions, 60 were indexed by the system and the rest are used to measure the precision and recall of the system. From the 64 questions to be answered, the FAQ module managed to answer correctly 48 questions, 15 had incorrect answer and only failed once to provide any answer. This gave us a precision and recall of 75% and 76%. The reason why it failed to correctly answer all questions was the problem of transitivity. As an example taking the following indexed questions, which are a reformulation of each other, could fail to identify the similar question “What could I see in Reims?”:

What tourist attractions are there in Reims?
What are the names of tourist attractions in Reims?
What attracts tourists in Reims?

Since the word “see” and “attractions” have no relation in wordnet this question reformulation identification fails. However if the question “What is worth seeing in Reims?” is added to the indexed questions it will then succeed because of the keyword “seeing” which has a root form of “see” (see+ing). This module will get more accurate as more questions are added to the index. Since no other QA systems have employed this technique comparing it with others was quite impossible.

The AE module as well scored quite a high precision and recall of 87.5% and 63.64% respectively. Evaluating this module involved indexing six comprehension texts and the use of 22 questions. From this questions 14 were answered correctly and 2 were incomplete answers. An answer was considered correct if the four top most results returned contain the answer. Analyzing these results the system failed to answer some questions because either it failed to identify key terms in text or otherwise the answer returned was incomplete. In other words, an incomplete answer is a text segment with a 250 bytes limit but where the answer to the question occurs outside the answer window. In our opinion the system competes well with the other QA systems appearing in the TREC-9. However since the TREC data was not available for us comparing its performance with other QA systems was not possible.

One major drawback of the system is the system response time. Answering questions that need to score several text segments from a huge number of documents is a lengthy process. Various attempts to speed things up were tried like changing from a DOM parser to a SAX parser, moving the entity name recognition during document indexing and spawning several threads working independently on each document. The latter effort has not been measurable on a single processor machine, though on a better hardware the system will surely perform better. Despite these efforts the POS (Part-Of-Speech) tagger and its dependency on XML data was seen as the major drawback for performance. Considering of rewriting certain modules to remove the need of XML data at this stage to achieve the same results was not seen as feasible. However these changes are outlined in the future enhancement section below.

6 Future Enhancements

The system can answer factual questions with a high degree of accuracy. Increasing this accuracy involves a better method for term expansion. Although this is implemented in the system by using Wordnet synonyms the expansion method was not seen very useful. A very effective method is to employ a Word Sense Disambiguation (WSD) method for the words so that it guides the expansion of terms and thus resulting in a better chance to retrieve interesting documents. Another effort is to remove the XML data for improving the system performance. One way to achieve this is either swapping the Lucene search engine with another that can perform indexing at passage/paragraph level or else modify the Lucene indexing method.

Other improvements that might be considered in the future is to go beyond answering factual questions and cater for more complex questions. This can involve adding a new module to the system that can handle reason questions by making use of a knowledge base and theorem provers. One drawback of this solution is the need of a domain dependent solution, which contradicts our main aim of an open-domain solution. Therefore research in this area on how to achieve an open-domain solution while using a knowledge base can be studied further and included with this system.
References

10. Ellen M. Voorhees and Dawn M. Tice, Building a Question Answering Test Collection, National Institute of Standards and Technology.
18. Ian H. Witten, Alistair Moffat and Timothy C. Bell, Managing Gigabytes, Morgan Kaufmann, 1999
19. Xin Li and Dan Roth, Learning Question Classifiers, Coling 2002.
Generic Chromosome Representation and Evaluation for Genetic Algorithms

Kristian Guillaumier
Department of Computer Science and AI,
University of Malta

Abstract. The past thirty years have seen a rapid growth in the popularity and use of Genetic Algorithms for searching for optimal or near-optimal solutions to optimisation problems. One of the reasons for their immense success is the fact that the principles governing the algorithm are simple enough to be appreciated and understood. The major differences between one Genetic Algorithm and another lie within the schemes used to represent chromosomes, the semantics of the genetic operators, and the measures used to evaluate their fitness. Yet, these very differences make Genetic Algorithms so complex to design and implement when opposed with most real-world optimisation problems. The truth is that the people faced with these types of optimisation problems are not necessarily computer scientists or machine learning experts. Indeed, these types of problems constantly appear in various non-computing disciplines ranging from biology to manufacturing and economics. In this report, we present a simple, yet powerful, high-level technique that can be used to describe the structure of chromosomes and how their fitness can be evaluated. The method is abstract enough to insulate the practitioner from all the implementation, design, and coding details usually associated with a Genetic Algorithm. Nonetheless, a wide array of optimisation problems ranging from the classical travelling salesman problem and the n-Queens problem to time-table scheduling and dynamic programs can be described.

1 Introduction – Setting the Scene

A basic Genetic Algorithm may be decomposed into the following steps:

− Create a starting population. Usually a set of random chromosomes are created.
− Repeat the following until some termination criterion is met:
  • Evaluate each chromosome using a fitness function.
  • Select pairs of chromosomes using some scheme such as random selection or fitness-biased methods.
  • Apply crossover on the pairs of chromosomes selected and mutation on individuals.
  • Create a new population by replacing a portion of the original population with the chromosomes ‘produced’ in the previous step.

The above algorithm may be implemented in any high-level programming language. However, in conventional implementations, most parameters, the fitness function, chromosome representation, and genetic operators are usually hard-coded. If the nature of the problem varies slightly or critical parameters change, the original code must be revised – sometimes substantially. Moreover, as already stated, the user may not be even computer literate and not prepared to deal with issues such as algorithm design, programming and debugging.

In this section we will present a simple lecture-timetabling problem and eventually show how the procedure can be made more abstract.
Suppose we wish to find the solution to a very simple time-timetabling problem where each of 5 lectures has to be assigned one of 3 timeslots. No lecture given by the same tutor may occur at the same time. Sample data may be found in the tables below.

<table>
<thead>
<tr>
<th>Lectures</th>
<th>Timeslots</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>Name</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
</tr>
<tr>
<td>1</td>
<td>CSA1</td>
</tr>
<tr>
<td>2</td>
<td>CSA2</td>
</tr>
<tr>
<td>3</td>
<td>CSA3</td>
</tr>
<tr>
<td>4</td>
<td>CSA4</td>
</tr>
<tr>
<td>5</td>
<td>CSA5</td>
</tr>
</tbody>
</table>

Our first task is to find a suitable chromosome representation. In such cases, a direct representation scheme may be used. We use a vector of symbols of length 5 (one per lecture) where the symbol at the $i^{th}$ position holds the timeslot assignment for the $i^{th}$ lecture. So, the following chromosome:

\[
< 3, 1, 3, 2, 1 >
\]

would be interpreted as the first lecture in timeslot 3, the second in timeslot 1, the third in timeslot 3, and so on. Once having found a suitable encoding scheme, we proceed by selecting our genetic operators. In order to keep things simple, we apply basic single point crossover and random mutation as shown in the following diagrams:

We finally decide on how chromosome fitness will be evaluated. In this case, the fitness function can be defined to return an integer value representing the number of times lectures given by the same tutor occur at the same time. Since the return value of the fitness function may be considered to be proportional to the severity of constraint violation, it may be interpreted as a penalty that has to be minimised. Once the chromosome representation, genetic operators, and the fitness function have been defined, our Genetic Algorithm can be implemented as outlined previously.

So far, we have considered a simple, yet perfectly realistic implementation of a Genetic Algorithm for lecture timetable-scheduling. However, it should be immediately apparent that once the nature of the problem changes, say by introducing rooms or soft-constraints, the whole program must be revised substantially. This is because all the effort has been hard-coded and hence cannot be easily extended.
2 Abstracting the Problem

Many optimisation problems, directly or indirectly depend on sets of static data (such as the Lectures and Timeslots tables) and certain relationships between them. In the lecture time-tableing problem we have seen earlier on, each row in the Lectures table must be associated with a, yet unknown, row in the Timeslots table. We call relationships that exist but are yet unknown, dynamic relationships. In view of this, we found it convenient to express such static data and relationships as a variation of database tables found in relational database management systems (RDBMSs). The following figure shows the Lecture and Timeslots tables, and the relationship between them.

Note: In the Lectures table, we have included a 4th column called TimeslotCode. This column holds values matching those in the Code column of the Timeslots table and serves to create the relationship between the two. In database terminology, the TimeslotCode column is called a foreign key column.

These tables and their relationships represent the structure of the problem. Eventually this structure will determine the chromosome representation. The search process then attempts to find optimal, or near-optimal, values for the dynamic relationships – the TimeslotCode column above. Once the Genetic Algorithm starts producing candidate values for the dynamic relationships, a combination of queries, similar in concept to the Structured Query Language (SQL) Select statement and conditional If-Then-Else statements are used to evaluate fitness.

Suppose the Genetic Algorithm yields the following candidate values for the TimeslotCode dynamic column:

<table>
<thead>
<tr>
<th>Lectures</th>
<th>Timeslots</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>Name</td>
</tr>
<tr>
<td>1</td>
<td>CSA1</td>
</tr>
<tr>
<td>2</td>
<td>CSA2</td>
</tr>
<tr>
<td>3</td>
<td>CSA3</td>
</tr>
<tr>
<td>4</td>
<td>CSA4</td>
</tr>
<tr>
<td>5</td>
<td>CSA5</td>
</tr>
</tbody>
</table>

The following SQL statement may be used to retrieve the tutor and timeslot description from the two tables:

```
SELECT Lectures.Tutor, Timeslots.Description
FROM Lectures, Timeslots
WHERE Lectures.TimeslotCode = Timeslots.Code
```
Returning:

<table>
<thead>
<tr>
<th>Lectures.Tutor</th>
<th>Timeslots.Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andrew White</td>
<td>Monday 15:00</td>
</tr>
<tr>
<td>John Green</td>
<td>Monday 08:00</td>
</tr>
<tr>
<td>Michael Brown</td>
<td>Monday 12:00</td>
</tr>
<tr>
<td>Mark Black</td>
<td>Monday 08:00</td>
</tr>
<tr>
<td>Andrew White</td>
<td>Monday 15:00</td>
</tr>
</tbody>
</table>

If we name the query above \textit{qryTutorTimeSlots}, we can use the following conditional statement to return a penalty based on how many lectures are occupied during the same timeslot.

\begin{verbatim}
If TRUE Then
   Return \{Number of tutors in same slot from qryTutorTimeSlots\} * 10000
Else
   Return 0;
\end{verbatim}

Clearly, if no tutors are busy during the same timeslot, the if statement will return 0 – no penalty. Otherwise the value returned will be directly proportional the number of times the constraint has been violated.

3 Conclusion

In the previous section we briefly demonstrated how tables may be used to describe the structure of an optimisation problem, and how combinations of SQL-like queries and conditional statements can express the components of a fitness function.

This technique has been successfully implemented as a high-level modelling language called OPML\footnote{Optimisation Problem Modelling Language} together with a general-purpose Genetic Algorithm-based runtime. A wide range of optimisation problems have been tackled including problems in linear programming, dynamic programming, lecture time-table scheduling, the travelling salesman problem, bin packing, and the n-Queens problem as test cases.

For a complete exposition the reader is referred to:

A Metrication and Comparison Framework for E-Commerce Systems

Mark Micallef
Department of Computer Science and AI,
University of Malta

1 Overview and Motivation

“We live in an era of ’e-everything’ ” [1]. Everywhere we look, we see people trying to fuse technology with every imaginable business concept in the hope of making more money. We have seen e-commerce, e-procurement, e-logistic, e-government, and e-banking to name a few. The late 1990’s saw technology stocks on the market inflate to astronomical levels as would-be entrepreneurs wrongly reasoned that this wonderful new technology would automatically catapult them to unimaginable riches. Thousands of dot-com startups emerged in the in the early months of the new millennium, thousands of dot-com startups crashed.

We know now that setting up an online venture is no guarantee of success. Business creativity and discipline also need to be applied. The business and technical aspects of an e-commerce venture will together contribute to its degree of success (or failure) and one cannot go without the other.

Now that we have years of case studies available, can we apply theory, technical knowledge and good old-fashioned hindsight to predict whether and e-commerce system will succeed or fail? Are we capable of accurately forecasting whether one venture will fare better than another? Can a metrication framework for e-commerce systems be developed to measure the quality attributes of e-commerce systems? If so, at which stage of the development of a system would such a framework be applicable?

2 Project Outline

The project will attempt to develop a metrication and methodology framework for measuring the quality of e-commerce systems. Although this is principally a technological degree, research also needs to be conducted into the business aspect of e-commerce systems since business considerations will also affect the quality an e-commerce venture. Therefore this project will analyse e-commerce applications from principally technological lense but will also take into account the business point of view.

2.1 The Technical Point of View

The technical aspect of this project will focus on existing and upcoming technologies that are available and can be used in the design and implementation of e-commerce systems. A number of questions will be asked. The following are a few that initially come to mind:
(a) Are there any particular project life-cycles that lend themselves to better quality e-commerce systems?
(b) At what particular stage (if any) of a project’s life-cycle is quality built in?
(c) What traditional quality attributes can be used to measure the quality of e-commerce systems? Are these enough? Can we create new quality attributes specific to e-commerce products?
(d) What technologies are used throughout the lifetime of an e-commerce application?
(e) How can these technologies affect the quality of such a venture?
(f) Can a way be found to compare e-commerce products based on the different technologies, methodologies, etc use during their lifetime?

These questions will obviously evolve throughout the course of the project as continuing research will undoubtedly uncover more unanswered questions.

2.2 The Business Point of View

Here different business models and their applicability to different kinds of ventures will be reviewed. Various questions need to be answered at this stage. These include (but are not limited to):

(a) Are business models for "conventional" business applicable to e-commerce ventures?
(b) Have any business models been developed specifically for e-commerce ventures?
(c) How can one map a business model being used in an existing traditional business to an e-commerce model?
(d) What business-aspect quality attributes exists for e-commerce systems and how can they be measured?
(e) Can we propose new business-aspect quality attributes for e-commerce systems?

The aim is to merge both aspects together towards the end of the project in order to come up with a complete set of quality attributes, metrics for measuring them and accompanying methodologies that encompass activities and processes over the whole spectrum from the purely business aspect to the purely technical.

3 Expected Activities

The project is envisaged to involve a mix of activities:

- Research into work that has already been done in this area
- User surveys to help establish what quality attributes users of e-commerce systems appreciate most
- Industry surveys to get a feel of what development methodologies are used by web development companies as well as their general outlook towards building quality into e-commerce systems.
- Development of a metrical framework to measure the quality of an e-commerce system
- Application of the metrical system to a life case study in order to test its validity
- Implementation of a software tool to demonstrate the metrics, methodologies and results developed throughout the course of the project.
References

WWW2: Second Generation Web – Semantic Web

Matthew Montebello
Department of Computer Science and AI,
University of Malta

Abstract. The World-Wide Web (WWW) has ever been evolving since its inception in 1989 not in regards to the wealth of knowledge that is deposited and stored on server machines scattered across the evergrowing network of online servers, better known as the Internet, but in regards to its functionality, potential use, and capabilities. In this paper we investigate this evolution with the aim to better understand the natural transition to the next generation WWW — the Semantic Web.

Overview

The Internet is the World-Wide network of computers that began in the 1970s as a communications network between American government defence organisations. In 1984 control over the network was handed to the private sector and by 1989 the WWW, often referred to as the web, was developed. The WWW, which came out of work by the British scientists Berners-Lee et al. [1] working at the European particle physics research establishment, CERN in Geneva, is the whole constellation of resources available on the Internet. The main idea is to merge the techniques of computer networking and hypertext into a powerful and easy to use global information system. Hypertext is text or graphics with links to further information, on the model of references in a scientific paper or cross-references in a dictionary. With electronic documents, these cross-references can be followed by a mouse-click, and in the case of the WWW, the staggering wealth of knowledge and experience, deposited and stored on server machines scattered across the Internet, is available to its on-line world.

When people access the web, they are either searching for specific information, or they are simply browsing, i.e. looking for something new or interesting (by a process often referred to as surfing). These two activities, searching and browsing, have associated applications over the WWW to access and help identify resources within the electronic documents respectively. The main difference between browsing and searching is that while no search statement or query is specified during the browsing activity, users explicitly express a search statement by submitting a query defining their information requirement to a web-search application, better known as a search engine. Links to potentially relevant information is presented by these search engines to the users, who can access and browse the information using a web browser.

In this paper we investigate the evolution from simple web-page browsing and surfing using plain Hypertext Markup Language (HTML) in the next section to slightly more complicated Extensible Markup Language (XML). A whole plethora of events characterised this path but a natural evolution justifies why the Semantic Web is the next sensible step within this sequence of technological development to maximise and enhance the use of the WWW. Topics will be discussed in this order:

- HTML and its limitations
- HTML+
- XML
- XML+
- Semantic Web
References

A Log Analysis based Intrusion Detection System for the
creation of a Specification Based Intrusion Prevention
System

Andre' Muscat
Department of Computer Science and AI,
University Of Malta

Abstract. We propose a novel Intrusion Prevention System (IPS) which would base its
knowledge and operation on a higher level of abstraction than the processing of the contents
of the network packets audit data themselves which is the source of data on which most
current and proposed Intrusion Detection Systems (IDS) base themselves on. We focus on
what is actually being asked of the system, and use that understanding together with research
on prediction based systems to build a specification based Intrusion Prevention System based
on the patterns extracted from higher level application or operating system logs.

1 Introduction

While the world of Intrusion Detection Systems (IDS) technologies and methodologies for detecting
attacks seem to move ahead at the speed of light, this is not quite the case. By observing a report
commissioned in 2000 by the Software Engineering Institute (SEI) [12] coupled with my experience
in the security field for over four years, one would be able to notice that the systems used for
IDS have not developed so much. Granted, features and functionality are being integrated into
products rapidly, however the methodologies used in detecting attacks did not adhere to such a
tight schedule. All detection mechanisms used to expose illicit activity on a network can be broken
down into two main categories which do not overlap. These are Pattern Matching and Anomaly
Detection.

Pattern Matching based IDS work by looking for patterns/signatures in the information data they
are processing in order to detect attacks. When a pattern in the data fits a signature the system
would issue an alert to the security administrator. As a direct result of this operational method,
pattern matching based solutions are generally effective at detecting known current intrusion meth-
ods, but quite ineffective against novel or new attacks until a signature for that new attack method
is released.

Anomaly Detection based IDS work by establishing a “normal” operating method of a system.
They create patterns of usage of that machine. When some action/activity happens outside of the
normal operational usage patterns, it would trigger an intrusion alert. Anomaly Based IDS, unlike
Pattern Matching IDS are generally quite good at detecting novel attacks but can have a tendency
to generate many false positives as a result of “off the normal” legitimate actions which a user of a
system might perform. From personal experience we came to think that unlike several marketing
campaigns claim, all IDSs on the market use some combination of the above methodologies for
detecting attacks using different types of data sources.

In this paper we will be analyzing the reasons why a security administrator of an organiza-
tion/networked system would require an Intrusion Detection System as part of their standard
collection of tools which are required for the proper daily operation of their administration and resource security enforcement job. However we will not stop there. After making a case on the importance of having an IDS and how they contribute to close “holes” inside a network setup, we will also proceed to analyze how totally unrelated (non IDS) tools, which a typical high end administrator uses in his daily task of administration and integrity keeping of his network can be used in order to create a system which based on IDS technology can actually be used to create an Intrusion Prevention System – an IDS with the capability to react based on intelligent information processing. This will allow the tool not only to detect an intruder but will also be able to stop the intruder point blank during the course of an attack.

We will be covering both common methods through which an attacker compromises an operating system for his own current and future needs, as well as analyze what is being done by security engineers and experts in order to learn how attacks are being made directly from the source, i.e. the attackers themselves without paying them or promising rewards. This system/method used is called a HoneyPot/HoneyNet [15] and is proving to be very useful to a security administrator who wants to be instantly aware of new attack techniques/variations of already existing attacks that are being used.

At the end of the paper we will also be mentioning a novel intrusion prevention method which is aimed at blocking a specific type of attack – buffer overflow attacks which is still in the research phase.

2 The Need For Securing Resources

Any computer user has heard the buzzword “security” and the need to protect the information contained on a private computer from prying eyes. The main problem to be replied in most if not all cases is “How can this be achieved ?”. From experience, a security aware administrator knows that the best solution is to deploy defenses on a network in a layered manner – not one defense mechanism, but many different types and at different levels addressing different monitoring needs. These defenses include the deployment of secure operating systems, the creation and deployment of security policies, vulnerability scanning, patching, access control and authentication, encryption, program wrappers, firewalls, intrusion detection, and intrusion response, and also disaster recovery software/procedures as well as proper configuration of the operating system features and services to support the required security policies. Although this list is quite long a security aware administrator would also be required to train the people making use of the system in order to ensure that the secure system remains secure. Irrespective how many systems/policies one would deploy all of them will be nullified by a user of the system who does not understand and follow the security practices required. For e.g. a user with an easily guessable password like “love”, name of wife, name of car etc can nullify all of the security measures implemented since an intruder would use those credentials to make use of the system resources. More information on the dangers of weak passwords and how they can be prevented/enforced by an operating system based security policy can be found at [13].

With the developing complexities and improvements of operating system, security software, security standards and the need of people to know what is going on their network, the need to provide more human readable and processed data was required. This data is the first point of reference a security administrator would run to in order to check out what is going on the network, on the first
suspicion of an intrusion/abnormal/not-desired activity. This data takes the form of logs, which are outputted from all of the above mentioned types of software which are used to enforce security and understand what happened on that system in order to correct the security measures on that PC and prevent such incidents in the future.

Given a multilayered type of network security deployment even in a moderately sized network of 20 computers would generate more information and logs which an administrator would find heavy difficulties in be able to keep up with in order to monitor the daily activity on the network. From our own personal experience in the field, these logs are typically only accessed depending on the problems/reports encountered as the days go by. However this means that if a suspicion is in place the attack may have already been done and executed, which in itself may be too late to protect against for this attack, however on a brighter side the information collected from this attack may be used to prevent future similar attacks.

Current IDSs might inform a user that an intrusion has occurred, however to find out how this was done, an administrator would still be required to delve into these operational/trace logs to look for some indication on what mis-configuration/vulnerability was used by the intruder. Even more worse, without any form of monitoring a user might have left some files/tools around the operating system purposely. These files may be used to hide the actions performed or even worse replace normal operating system files containing custom backdoors in order to enable easier access on future attacks without being detected. Unfortunately, without total monitoring/detailed protections, once a system is compromised (after a successful intrusion), without a proper prevention system, the entire system would have to be re-installed from scratch in order to ensure the integrity of a computer system. This is the main concept used by RootKits to take control of a system. These kits are developed specifically to operate at the kernel level rendering them fully capable access everything and everywhere while being able to hide the operations of an attacker from the unwitting security administrator. RootKits replace central operating system files giving the impression that nothing changed, to the naked eye. RootKits can either take the form of Trojans attaching themselves to files or by totally replacing the original core operating system files with custom made system files with custom code specifically designed to hide illicit activity on that system by the attacker. More information on RootKits can be found at [11] and [12]. Special file system level integrity monitoring and Trojan detection tools can be used to be informed when such sensitive files are changed. Tools such as Tripwire [8] are used for such file level intrusion change detections.

Even well administered networks are vulnerable to attack. Recent work in network security has focused on the fact that combinations of exploits are typical means by which an attacker breaks into a network. [5] Without multiple layers of security software deployed on a system, that very same system would be just as strong as its weakest point of entry. This creates problems since there are automated tools through which attackers would simply supply a machine IP, let it operate and let it report the weaknesses of that system. Based on those weakness reports the attacker would be able to devise his next steps of attack. A growing tool in the open source community is working on a project whose output is a tool named Nessus [9] which although designed for administrators to find the “weakest link” in the security infrastructure of a system, can be itself used by an attacker to devise his attack pattern in order to go by undetected.

3 NIDS vs. HIDS

In the field of detection of unwanted intruders on a computer network system, there are two main types of Intrusion Detection Systems (IDS) which act on different sets of data. Network Intrusion Detection Systems (NIDS) operate on the network packets audit data being received at the network point of a machine, while Host Based Intrusion Detection Systems (HIDS) work by monitoring the actual applications behavior on the host and its interactions with the underlying operating system.
It is a fact that in the past data contained within a packet received at the network level was easier to access, decode and process. This resulted in a concentration of research and development of several IDS systems based on the processing of information contained in network data packets. The literature field also seemed to concentrate most on this type of data. However with the movement of high speed Ethernet cards from 100Mbs to 1GBps, coupled with the encryption of the data contained in the packets via the use of technologies such as IPSEC/SSH, it has become increasingly difficult for NIDS to keep up both the precision at which they can detect intrusions as well as guarantee the detection of attacks without a high rate of false positives.

In this research task, we are giving Host Based IDS the attention it needs in order to use current technologies to process as much of the information we may already have already at hand in order to build patterns and systems, which can actually not only detect, but also prevent attack methods.

In HIDS, the most common approach taken is to monitor relevant interactions between an application and an operating system by monitoring the system calls on that operating system. This resulted in the creation of systems which are based once again on monitoring a pattern of calls rather than the real logical operations that an attacker would be thinking in order to gain access to a system. Such a level of analysis also makes it easier for an attacker to hide the tracks of the attack by mimicking pause periods and fake system calls to hinder system call based HIDS by applications.

4 Data Mining Applications

Typically most IDS technologies require the use of pattern constructs which are used to either define attacks/abnormalities. Whether we are working on an IDS or an Intrusion Prevention System (IPS), we still require the creation of patterns via manual Knowledge Engineering from a network/security expert, or the use and application of modern methodologies to process all of the large amounts of information, in order to extract patterns which can be monitored and adapted by the knowledge engineers. Given the complexities of today’s network environments, needs and the sophistication of the increasingly hostile attacks the knowledge constructed based outputted from the knowledge engineer only is often limited and unreliable [1]. Data Mining applications is a field which is directly addressed towards large data set information grouping and searching. Data mining is an area which can offer large opportunities to the refining of the pattern matching rules used by IDS.

“Data mining tasks and algorithms refer to the essential procedure where intelligent methods are applied to extract useful information patterns from huge data sets. There are many data mining tasks such as clustering, classification, regression, content retrieval and visualization etc. Each task can be thought as a particular kind of problem to be solved by a data mining algorithm. Generally there are many different algorithms which could serve the purpose of the same task. Meanwhile some algorithms can be applied to different tasks” [16].

5 Pre-Processing

As mentioned earlier in the paper, we are proposing the creation of a specification based IPS based on the patterns which are extracted from the various forms of logs which are outputted from all of the various types of software mentioned earlier on. However this is not as simple as it might sound. Most of the logs which are output from the various applications are proprietary and not conforming to a specific standard of any form. This means that the information contained is essentially different from one log to another. What makes this task even more complicated is that different logs will
offer different views to the system. In essence this task involved the analysis of various types of log files in order to help find commonalities and also group different type of logs in order to help create a framework which would allow the attainment of our aims.

Another aspect which is involved in log analysis is the inherent noise, missing values and inconsistent data in the actual information contained and written in the logs. It is a fact that all attackers know that all applications are vulnerable in some form or another. During research we also came across situations whereby attackers found and used methods which by supplying say a firewall with a particular type of data (in UNICODE format) they could literally crash the firewall and hence have absolute access to the system, as well as hide details of their activity since the firewall was not logging the information coming from the source of the attack. A log analyzer suffers from that very same danger. The importance to transform and process the data supplied to the log analyzer in a standard format is a key process to the standardization of the logs to be processed as well as offer an extra layer of protection before actual processing of the data, for e.g. Microsoft use their Event Viewer for log output, third party tools their own etc. A standard by a security company named NETIQ proposes a format named the WebTrends Enhanced Log Format (WELF) format which proposes a standard which is most ideal for large scale text based logs. With this format we can transform practically every log entry/system call equivalent to a standard format on which operations will take place. In addition real world data sets tend to be too large and some multi dimensional. Therefore we need data cleaning to remove noise, data reduction to reduce the dimensionality and complexity of the data and data transformation to convert the data into suitable form for mining etc.

6 Common Pitfalls

The use of anomaly detection in practice is hampered by a high rate of false alarms. Specification based techniques have been shown to produce a low rate of false alarms, but are not as effective as anomaly detection in detecting novel attacks. While detection of new novel attacks can be an issue, we are working on a system which will prevent known future attacks which are based on common methods. As an example in windows operating system environments, it is common for attacks to originate from the usage of particular sensitive operating system files – cmd.exe, ftp.exe etc. Indeed these files can be protected by basic file system level protection, however in large systems, file system based protection is not always possible and may require another type of level of protection. The rationale is that although an attacker may find a common operating system weakness, an extra level of protection based on specifications coming from previous analysis steps can be used to stop the attacker even if the underlying operating system defenses have been beaten.

7 Modern Techniques Used For Knowledge Discovery

To collect information on the way attackers are using system vulnerabilities, organizations and research groups are making use of what is called honey pots and empty systems whose function is to react like a normal operating system, however while giving the impression to the attacker that they got access to a legitimate system, the attacker is actually revealing his methods, tools and tricks to a highly monitored and specialized environment based exactly to root out the modern attacks. These systems are what are known as Honey Pots. The information collected from the honey pots and nets can be then combined, pre-processed, transformed and processed in order to devise specifications which can be then fed into our IPS in order to identify activities on a high
level logistical later and prevent by reacting to/redirecting attackers as we may need in order to subvert attacks. The simplest form of reaction is to close a connection, however unless the IPS is capable of detecting complex logistical issues and be able to think on a high level like a user, it will always be open to further attacks and weaknesses.

8 Being Realistic

For the development of an IPS we have to be able to assume that an attacker can silently take control of an application/operating system itself without being detected, via some form of social engineering paradigm or system vulnerability/exploit. Our system must also be able to cater for the unknown. There are actual penetration methods which can leave no trace in the system call trace especially if no logging is enabled. For instance exploiting a buffer overflow attack involves only a change in the control flow of the program, but does not itself cause any system calls to be invoked and thus no system call based IDS can detect the buffer overrun itself. Since no system call is made even the operating system would not report anything. Indeed an IDS which monitors the integrity of the code segment of an application would be able to report such an attack, but would be difficult to prevent it. Work is being done to see whether a way can be found to block this type of attack as soon as the MemCpy Command is used.

Contrary to popular belief there is little to be done against such attacks. Fortunately there is little harm that an attacker can do to the rest of the system without executing any system call. However this proves the point that different IDSs are required at different levels. An IDS which checks for the integrity of the code segment of a program can be sufficient to detect such an attack but not block it/prevent it.

In order to address such attacks and also prevent them were possible we are thinking of applying the same concept used in the processing of the logs to treat the parameters passed to a system call as a field entry in the logs. Like that we may be able to perform checks on the parameters of the call and hence also start preventing buffer overflow attacks even the actual call is executed by monitoring the request. This part is still in research and treating it as a section for Future Research.

9 System Implementation

As a result we need to find patters in order to model and understand what a user is effectively using and doing rather monitor only the patter of operating system calls. Based on the patterns which are generated either by Knowledge Engineering or patterns resulting from Data Mining Techniques we can build a Finite State Machine Based System through which we can determine/predict the next state based on a number of previous states and if our prediction comes to reality also decide to block the operation from taking place.

\[
\text{Action} \quad (E0, E1, E2, E3) \rightarrow (E1, E2, E3, E4)
\]

By monitoring a sequence of operations one can already predict the next operation based on past experience. The real trick in our proposal is to use the past experience patterns not to predict but to know what is going on on a system secured by our final system. This would allow instant intrusion prevention if our IDS is intelligent enough to understand and react at levels which are higher than operating system calls.
10 Conclusion

In these couple of pages we tried to present an outline of our work, however due to space limitations we had to skip some details which might have been more illuminating to the shady areas of the most discussed but yet unknown and mysterious field that is the security section of the computing world. The research covers various technologies used for security, intrusion detection, prevention, as well as systems and algorithms used by the above mentioned technologies including automation data mining systems and algorithms as well as the development of a framework which is operating system independent to be used for intrusion prevention as a protection against the attackers of tomorrow.

References

15. The Honeynet Project, 2001. Addison-Wesley. Know Your Enemy,
16. Tao Li, Qi Li, Shenghuo Zhu, Mitsunori Ogihara, 2003. A Survey on Wavelet Applications in Data Mining.
1 Introduction

The concept of Web Services has, throughout the last few years, become one of the most discussed in the academic, as well as the business world. Many hail it as a revolutionary concept, while others look at it in a sceptic manner. The fact remains that the developers of the most diffused programming languages are giving great importance to integrating and supporting the creation and utilisation of Web Services.

The main advantage of Web Services is that they are based completely on XML, which gives them a very high degree of flexibility and, above all, platform independence. Systems written using one particular language, can transparently access exposed services on other systems written using different languages.

This particular feature of Web Services puts them in an ideal position to be utilised as the driving force behind distributed systems. The Internet is a jungle of computers, each with their particular features, operating system and hardware platform. Such a scenario makes it very difficult to create a system, distributed over various machines, which easily adapts to such an environment and communicates with other machines in this environment. Using Web Services, this problem is naturally solved.

This project aims at analysing the ways in which Web Services may be used to enhance distributed systems over the Internet framework. There are various issues which should be considered, including the software required and the security of data being transmitted.

2 Scenarios

This project will focus on two scenarios which can be easily applied to real-life situations. The final product will be a demonstration of each of these scenarios, applied to a particular business area.

2.1 Scenario 1: Web Services as Notification and Control Tools

In this simple scenario, Web Services are used as a remote control and notification tool for subscribers to the particular system. A small application located on the client’s machine will poll the central system and retrieve any new notifications. The client will then be allowed to handle the notifications either through the client system, or via a web-based interface. Figure 1 illustrates the concept.
Fig. 1. Web Services as Notification and Control Tools

When a new request is posted, the system automatically notifies the interested client either via email, or through the web service interface, as chosen.

Web Service Client queries at regular, short intervals main system to retrieve new pending requests, or updates the central system with all its contents.
Fig. 2. Web Services as Integration Tools
2.2 Scenario 2: Web Services as Integration Tools

This more complex scenario puts Web Services as the main tool for remote integration and control. A central system is used to collect and disseminate information to and from various remote systems, integrating and displaying them as a single entity. This system allows third parties to utilise legacy systems for managing their information, while the distributed software node interfaces with such a system, extracting, translating and transmitting the information either automatically or on request from the central system. Web Services may be used in various degrees, depending on the requirements of the system, however, the main issue is whether the distributed node (as opposed to the central system) should either have the role of a client, or that of a server. The decision depends on whether such node should simply serve as a ‘dumb’ terminal which simply collects information and transmits it to the central system, or whether it should hold the information locally and only transmit the necessary details on request by the central system. The various aspects of this issue will be dealt with throughout this project, outlining the consequences, advantages and disadvantages entailed by each solution.

References

3. Web Services, the New Generation of Application Integration — Data Junction Corporation.
5. Web Services architecture overview, the next stage of evolution for e-business — IBM Web Services Architecture Team (karlgott@us.ibm.com) http://www-106.ibm.com/developerworks/web/library/w-ovr/.
Computer-Aided Verification: How to Trust a Machine with Your Life

Gordon J. Pace
Department of Computer Science and AI, University of Malta

Abstract. Mathematical predictive analysis of the behaviour of circuits and computer programs is a core problem in computer science. Research in formal verification and semantics of programming languages has been an active field for a number of decades, but it was only through techniques developed over these past twenty years that they have been scaled up to work on non-trivial case-studies. This report gives an overview of a number of computer-aided formal verification areas I have been working on over these past couple of years in such a way to be accessible to computer scientists in other disciplines. Brief mention is made of problems in these areas I am actively working on. It does not purport to be an overview of the whole field of computer-aided formal verification or a detailed technical account of my research.

1 Introduction and a Brief History

Today, software and hardware are used to control all sorts of devices — from washing machines and microwave ovens to braking systems in cars, medical devices and nuclear power stations. Engineering practice, as evolved over the centuries, dictates guidelines to follow when designing bridges and towers to guarantee that they do not fall. However, with software and digital hardware similar techniques fail to work for a number of reasons:

– A problem of scale: the techniques fail to scale up to the level of complexity found in software and hardware;
– Discrete vs continuous: the digital nature of computing systems precludes the effective use of algebraic laws in design which usually work well on problems with continuous variables;
– Discontinuity: thorough testing is nigh on impossible on discontinuous systems\(^1\).

The more software and complex hardware is used in safety-critical systems, the more worrying this lack becomes.

Since the 1960s, techniques have been developed to enable mathematical reasoning about programs. Mathematical models of digital circuits had been around even before. Such models are obviously necessary to be able to prove properties of programs. However, the complexity of proofs using these models is formidable. The formal verification of a five-line algorithm can easily run into pages of dense mathematical formulae. The question ‘Is the program correct?’ is simply transformed into another: ‘Is the proof correct?’ This problem was addressed by writing computer proof checkers and proof editors. These tools allowed a proof script to be checked by a machine. Checking a proof for

\(^1\) On a continuous domain problem, we know that if at a certain point the behaviour is correct, that it should also be so in the immediate neighbourhood of the point.
correctness turns out to be a computationally simple problem, and it is only the generation of the proof which is difficult. In 1971, Cook proved the boolean satisfiability problem to be \textit{NP-Complete} [10]. Writing a program to calculate a proof in polynomial time is one way of showing that the satisfiability problem lies in \textit{P} and thus that \textit{P}=\textit{NP}. This disillusioned those who hoped to add modules to theorem provers to generate proofs automatically\textsuperscript{2}.

In the late eighties and early nineties, researchers started looking at finite state systems whose states can be exhaustively enumerated. Typical examples of such systems are digital circuits, programs which do not use dynamic features, and systems which use a bound number of resources. The catch is that the number of states is exponential with respect to the length of its symbolic description. However, if we look at the \textit{reachable} states in the system, it turns out that in certain cases we can manage to (i) enumerate all the reachable states, verifying the correctness of the system, or (ii) enumerate sufficiently many reachable states until a ‘bad’ scenario is unveiled. This led to a surge of interest in \textit{model-checking} algorithms which verify a system automatically by systematically exploring its state space. In practice, applying such techniques blindly works only on toy examples.

Although it is still not known whether a polynomial complexity satisfiability algorithm exists, there do exist algorithms which are worst case exponential, but work reasonably efficiently on most typical cases. Such algorithms are usually symbolic in nature, in that they try to encode sets of different boolean variable assignments using a symbolic representation, thus allowing the description of large sets using limited space. These algorithms have been used in model-checking tools, pushing the limits of these tools to work on larger systems. This is usually called \textit{symbolic model-checking}, in contrast with \textit{enumerative model-checking} techniques already mentioned. Although symbolic model-checking may seem to be superior, it is largely a question of the problem domain as to which type of model-checking is more effective. In cases where large chunks of the state-space are not reachable, enumerative techniques may actually work better\textsuperscript{3}.

In the meantime, \textit{abstract interpretation} techniques also started being used, where the user gives information on how to abstract the system under examination, thus reducing the size of the state space. Abstraction can also used to reduce infinite state systems to finite state ones. Abstract interpretation algorithms which try to calculate an appropriate abstraction algorithm have also been proposed, and such algorithms can enable the automatic verification of systems of up to $10^{150}$ or more states\textsuperscript{4}.

Despite this seemingly huge number, this limit still falls well below the size of complex circuits currently being developed. However, the cost incurred by Intel when a bug was found on its Pentium chip, and the danger of a law-suit if, or rather when, a system failure directly leads to loss of life (or major financial losses) have driven hardware design companies to use formal verification techniques. In fact, a number of commercial hardware verification software packages have now been on the market for a number of years and most microprocessors and complex chips are partially verified before going on the market.

Software verification is much more complex than hardware verification. Strong reduction and abstraction techniques are required to be able to do anything automatically. However, limited success

\textsuperscript{2} In most programming languages (which may use features such as unbounded integers, unbounded length strings and dynamic lists) the problem is even worse. Gödel’s incompleteness theorem guarantees that no algorithm to decide the correctness of a program can exist.

\textsuperscript{3} For example, in the case of concurrent programs and communication protocols, the state of the system would also contain the program counters along the different threads. Most program counter settings are impossible to reach, making large partitions of the state-space inaccessible. In these cases, enumerative techniques tend to be more effective.

\textsuperscript{4} This figure is obviously dependant on the system in question. However, it is interesting to compare this to the typical limits of $10^7$ states for enumerative model-checking and $10^{150}$ for symbolic model-checking using no abstraction.
has been demonstrated on restricted domain problems, such as memory leak analysis and device driver verification. A number of commercial tools which use static analysis and verification techniques to guide programmers have just started appearing on the market.

Despite its short history, model-checking boasts a vast literature. A good starting point to the field is Edmund Clarke’s book [8] which gives an overview of model-checking at the time of its writing.

The rest of the report will outline a number of research areas on which I have been working over these past few years, and identifies the problems in which I am still working on at the moment.

2 Model Reduction Techniques

In enumerative model-checking, a major problem is that the composition of tractable systems may turn out to be intractable. Composition operators which may result in state-space explosion include asynchronous composition and partially synchronous composition, where some ports are synchronized, while the others are left to work independently. Unfortunately, these operators appear in various systems such as communication protocols, and thus cannot be avoided.

One solution proposed was to generate the individual elements of a composed system, and minimize the finite state machines as it composes them [11]. Systems are described in terms of a composition expression which states how the basic nodes of the system are composed together. A simple example of such an expression is:

\[
\text{receiver} \mid\mid \text{hide sel in ((sender1} \mid\mid \text{sender2)} \mid\mid \text{sel arbitrator)}
\]

where \(\mid\mid\) is synchronous composition, \(\mid\mid\) is asynchronous composition, and \(\mid\mid_P\) composes together two systems synchronizing all communication over ports listed in \(P\) but communicating asynchronously an all other ports. \(\text{hide P in S}\) makes ports \(P\) inaccessible outside \(S\). \(\text{sender1, sender2, arbitrator}\) and \(\text{receiver}\) are finite state machines. The algorithm would minimize these machines, compose them together and minimizes again, as it moves up the expression tree. In practice this algorithm significantly improves the performance of the finite state machine generation.

However, one problem remains. Quite regularly, the individual finite state machines allow for complex behaviour which will be simplified when composed with another machine in the system. In the example we have previously given, one could imagine complex senders which can emit data in any order. On the other hand, the receiver only accepts a number followed by a character. Combining \(\text{sender1}\) and \(\text{sender2}\) will result in an enormous automaton, but most of the states and transitions will be discarded when we later compose with \(\text{receiver}\). In practice we may not be able to generate the intermediate automaton due to its size, even if the top level automaton may be quite small. This and other similar problems severely limit this approach.

One solution to this problem was proposed by Krimm et al [13] where they propose a solution which allows us to reduce an automaton with the knowledge that it will be later composed with another node (called the interface). In the example we gave, \(\text{sender1}\) and \(\text{sender2}\) can be reduced by using \(\text{receiver}\) as an interface. This technique has allowed the verification of substantially larger systems than was possible before.

Given an automaton with \(n_1\) states, and an interface with \(n_2\) states, the algorithm proposed in [13] guarantees that the result will have less than \(n_1\) states, but requires space of the order \(O(n_1 \times n_2)\) to generate the reduced automaton. If the receiver in our example was too large, reduction may be impossible. To circumvent the problem, either the user would have to come up with a simplified interface (the algorithm makes sure that the interface is correct, so there is no risk of giving a
wrong interface) or one can try to generate a smaller version of the interface automatically to use. The problem of generating the smaller version of the interface automatically, can be expressed in terms of the language recognised by automata:

Given an automaton $M$, we require a new automaton $M'$ such that (i) $M'$ does not have more states than $M$ and (ii) the language generated by $M'$ is larger than that generate by $M$: $L(M) \subseteq L(M')$.

The problem has two trivial solutions: taking $M' = M$, or defining $M'$ to be the automaton made up of a single state and which can perform any action in $M$. However, we would like a solution which allows an effective reduction when used as an interface. I have been looking into different ways of generating an effective automaton $M'$. The applications of this are numerous and a number of case studies which use interfaces for verification are available, and can be extended to the effectiveness of the alternative interface generation.

Another reduction technique, based on composition expressions has been identified in Pace et al [15]. It has been shown to be quite effective when reducing machines with internal actions which are not visible to the outside user. The interaction between and combination of these two reduction algorithms which use composition expressions has not yet been studied.

### 3 Model-Checking a Hardware Compiler

There are two essentially different ways of describing hardware. One way is using structural descriptions, where the designer indicates what components should be used and how they should be connected. Designing hardware at the structural level can be rather tedious and time consuming. Sometimes, one affords to exchange speed or size of a circuit for the ability to design a circuit by describing its behaviour at a higher level of abstraction which can then be automatically compiled down to structural hardware. This way of describing circuit is usually called a synthesisable behavioural description. Behavioural descriptions are also often used to describe the specification of a circuit.

Claessen et al [5] have used a technique from the programming language community, called embedded languages, to present a framework to merge structural and behavioural hardware descriptions. An embedded description language is realised by means of a library in an already existing programming language, called the host language. This library provides the syntax and semantics of the embedded language by exporting function names and implementations.

The basic embedded language used is Lava [7]. Lava is a structural hardware description language embedded in the functional programming language Haskell [16]. From hardware descriptions in Lava, EDIF netlist descriptions can be automatically generated, for example to implement the described circuit on a Field Programmable Gate Array (FPGA).

If one looks at the compiler description in Haskell, the code is short and can be easily understood. Consider a regular expression compiler: Given a regular expression, the compiler will produce a circuit which once reset, will try to match the inputs to the regular expression.

```haskell
compile :: RegularExpression -> Signal -> Signal
```

5 These are to be distinguished from behavioural descriptions (as used in industrial HDLs such as Verilog and VHDL) which are used to describe the functionality of a circuit, but are do not necessarily have a hardware counterpart.
For example, the circuit compiled from the regular expression (using Haskell syntax) is $\text{Wire } a \oplus: \text{Wire } b$, will have one reset input, and one output wire. $\text{Wire } a$ is interpreted as ‘wire $a$ is high for one time unit’ (corresponding to the normal regular expression $a$), and the plus symbol is language union in regular expressions. The resulting circuit would thus output high if and only if it has just been reset and, either wire $a$ or wire $b$ (or both) carry a high signal. Compilation of language union would simply by expressed as:

$$
\text{compile } (e \oplus: f) \text{ reset } = \text{or2 } (\text{match}_e, \text{match}_f)
$$

where

$$
\text{match}_e = \text{compile } e \text{ reset}
$$

$$
\text{match}_f = \text{compile } f \text{ reset}
$$

Note that each language operator would simply add a finite amount of new circuit components. Now assume that we would like to prove a compiler invariant\textsuperscript{6} $\text{inv}$. This can be proved using structural induction by showing that expressions such as:

$$(\text{inv } (\text{reset}_1, \text{match}_1) \land \text{inv } (\text{reset}_2, \text{match}_2) \land \text{plus_circuit}((\text{reset}_1,\text{match}_1),(\text{reset}_2,\text{match}_2),(\text{reset},\text{match})) ) \Rightarrow \text{inv } (\text{reset},\text{match})$$

This is actual Lava code, where $\land$ is conjunction and $\Rightarrow$ implication. The function plus_circuit simply ensures that the subcircuits are related according to the circuitry introduced for compiling $\oplus:$.  

$$
\text{plus_circuit}((\text{reset}_1,\text{match}_1),(\text{reset}_2,\text{match}_2),(\text{reset},\text{match})) =
$$

$$
(\text{reset}_1 \leftrightarrow \text{reset})
$$

$$
\land (\text{reset}_2 \leftrightarrow \text{reset})
$$

$$
\land (\text{match } \leftrightarrow \text{or2} (\text{match}_1, \text{match}_2))
$$

All this can be elegantly expressed in terms of embedded languages. Note that the cases of the structural induction are of a finite nature and can thus be model-checked. This gives an excellent platform to experiment with and verify compiler invariants, and thus hardware compiler correctness.

In unpublished experiments I have carried out with Koen Claessen, we have managed to prove that a regular expression compiler satisfies standard regular expression axioms. I am currently trying to extend this work for compilers of more complex languages such as Esterel \cite{esterel}.

\textbf{4 Hybrid Systems}

In real life, digital systems interact with analogue ones. While programs, and digital circuits can be fully analysed in terms of boolean values, a large class of (classical) engineering systems can only be modelled as a set of differential equations. Typical systems are expressed as a set of equations, defining the rate of change (with respect to time) of the variables which describe the system. Quite frequently, however, one needs to look at the interaction between continuous and discrete systems.

\textsuperscript{6} In this context, a compiler invariant is a property relating the inputs and outputs of a circuit produced by the compiler whatever the compiled program.
A typical example is a thermostat controlling a heating element. The equation defining the temperature can be defined in terms of a case equation:

\[
\dot{T} = \begin{cases} 
- l(T) & \text{if } T \geq 28 \\
H - l(T) & \text{if } T < 28
\end{cases}
\]

Where \( T \) is the temperature, and \( \dot{T} \) is the rate of change of the temperature with respect to time. \( l(T) \) is the rate of heat loss at temperature \( T \), and \( H \) is the heating rate of the thermostat (which turns on when the temperature falls below 28). Note that the system can be in either of two different modes or states, one with the thermostat turned on and the other with the thermostat turned off.

In such systems, discrete variables determine the system states which, in turn, determine the differential equations which describe the behaviour of the system variables. Such a system is called a hydrid system. Typically, hybrid systems are described using timed automata [1]. The thermostat example can be seen expressed as a (simplified) timed automaton in the following diagram:

\[
\begin{align*}
T > 28, \\
\dot{T} &= H - l(T) \\
T < 28, \\
\dot{T} &= -l(T)
\end{align*}
\]

As already noted, analysis of such systems is extremely difficult, and is in general undecidable. There is substantial research attempting to identify subproblems which are decidable. One such class of problems are Simple Planar Differentiable Inclusions (SPDIs) as identified by Asarin et al [2]. SPDIs are a subclass of hybrid systems with only two continuous variables. The best way to understand this subclass is to look at their visual representation.

An SPDI can be represented as a finite set of adjacent polygons on a plane. With each polygon we associate a differential inclusion (a pair of vectors which define the range of the rate of change of \( y \) with respect to \( x \)).

A typical example to illustrate SPDIs is the swimmer problem which depicts the motion of a swimmer in a whirlpool:

The differential inclusions define the directions in which the swimmer may swim when in a particular region (polygon). When the swimmer reaches the edge of a region, her dynamics change to match the inclusion in the new region. The figure also includes a path which may be taken by the swimmer under the SPDI constraints.
The kind of question one would typically want to ask of such a system is a reachability question. If the swimmer were to start in a certain location, can she end up in a dangerous location? In [2], Asarin et al have proved the correctness of an algorithm to decide this question for SPDIs. The algorithm depends upon the identification of the concept of meta-paths each of which describes a family of paths within the SPDI (abstracting away the number of times loops are taken). Meta-paths enjoy a number of interesting properties:

1. Given any two polygon edges in the SPDI, we can find a finite number of meta-paths such that any path in the SPDI starting on the first and finishing on the second edge is an element of at least one of the meta-paths.
2. This set of meta-paths can be algorithmically generated.
3. Given a meta-path, we can decide whether there is a feasible path obeying the differential inclusions in the SPDI along that meta-path.

These three properties guarantee the decidability of the reachability problem. The algorithm was implemented in [3]. The main shortcoming of the algorithm is that it searches the graph in an inherently depth-first manner. Even though a short counter-example may exist, the algorithm might have to try very long traces before finding the short counter-example. This also means, that unless an exhaustive search is done, we cannot guarantee that the counter-example found is the shortest one possible (a desirable thing since it would then simplify debugging of the system).

In (the as yet unpublished) [14], we have identified a breadth-first search solution to the problem. The algorithm is still based on the idea of meta-paths, but information is stored in a combination of enumerative and symbolic techniques so as to be able to express the solution in terms of a more standard reachability algorithm:

```
reached := \{ start_edge \};
repeat
  old_reached := reached;
  reached := reached \cup one-step(reached);
  if (not disjoint(reached, finish_edge))
    then exit with counter-example;
until (reached \approx old_reached);
report 'No counter-example';
```

The algorithm starts with the start_edge and adds other edges as they are reachable in one step (a path from one edge to another in the same region), until either nothing new is added (with certain provisos) or the finish_edge is reached giving a counter-example.

This algorithm is still to be implemented and compared to the depth-first version. Furthermore, the techniques used in our new algorithm (in particular the conditions required for the proof) indicate that a class of automata, more general than SPDIs, amenable to this technique can be identified. Work still needs to be done to identify this class and prove the generalised results.

5 Conclusions

In this report I have tried to give a brief outline of the history and relevance of formal verification techniques, to set the scene for some of my recent research and its current and future directions. At the risk of sounding like a collection of paper introductions and conclusions sections, not much detail is given in any of the research topics. Having only recently returned to the University of Malta, I hope that this report defines some research areas in which I am interested, possibly leading to fruitful collaboration with others in the department.
References

Finite State Analysis of Prepositional Phrases in Maltese

Michael Rosner
Department of Computer Science and AI, University of Malta

Abstract. We address the general problem faced by designers of computational lexica: that of relating surface forms to underlying lexical forms through the vehicle of a precise linguistic description expressed in a suitable formalism. For such a description to be useful, it must be possible for the relation to be computable: given a surface element, we need to compute the corresponding lexical element or elements, and vice versa. Below we concentrate upon the description of a minimal example: prepositional phrases, a reasonably well-defined and compact subset of Maltese surface phenomena that exemplifies many of the difficulties that are typical of Semitic languages.

1 Introduction

The work reported here is carried out under the general umbrella of Maltilex, a research programme supported at the University of Malta that aims to develop algorithmic and data resources for the Maltese Language (see Rosner et. al [8, 9]). Most of the effort so far has been directed towards the goal of automatically structuring lexical entries acquired from corpora. LST is the name given to the structuring technique, which is the subject of a recent MSc dissertation by Angelo Dalli (see [5] for a detailed description of the technique).

LST is essentially a statistical process that has no knowledge of the internal structure of words. This is both a blessing and a curse. The great advantage is that linguistic expertise is not necessary. The disadvantage is that if you are in possession of linguistic expertise, it may be difficult or impossible to incorporate it into the system.

Another major goal of the project is therefore to find ways of incorporating certain kinds of linguistic knowledge (morpholotactic knowledge in particular). For this to be possible we must (a) improve our understanding of morpho-grammatical aspects of the language (b) construct computational models and (c) develop means of biasing LST to take account of them.

This draft represents a first step in that direction and presents a description that deals with a well-defined part of the problem of mapping between underlying lexical strings and orthographic words (i.e. surface strings appearing between spaces).

Here we restrict our attention to simple prepositional phrases which are made up of a preposition, an (optional) article, and a noun. The first part of the abstract presents the linguistic facts. This is followed by a computable description using xfst [3], a finite state notation and compiler that has been developed at Xerox, and a brief presentation of the grammar.

We hope to extend this draft into a full paper in due course.
2 The Maltese Language Alphabet

The Maltese alphabet comprises 24 consonants and 6 vowels as shown below:

<table>
<thead>
<tr>
<th>consonants</th>
<th>vowels</th>
</tr>
</thead>
<tbody>
<tr>
<td>b c d f</td>
<td>a</td>
</tr>
<tr>
<td>ġ g gh h</td>
<td>ħ</td>
</tr>
<tr>
<td>h j k l</td>
<td>ħ</td>
</tr>
<tr>
<td>m n p q</td>
<td>ie</td>
</tr>
<tr>
<td>r s t v</td>
<td>o</td>
</tr>
<tr>
<td>w x ħ z</td>
<td>u</td>
</tr>
</tbody>
</table>

2.1 The Definite Article in Maltese

Definiteness in Maltese is expressed by preposing the definite article to the noun to form a single word. The unmarked form of the article takes the orthographic form l- (including the dash), e.g. l-ittra (the letter), but this can change depending on the surrounding morpho-phonological environment.

The two main phonological phenomena giving rise to these changes are referred to (see [6]) as /i/-epenthesis and consonant assimilation.

/i/-epenthesis involves insertion of the vowel i and serves to aid pronunciation. When this occurs purely as a result of characteristics of the noun alone, it is called inner epenthesis, whilst if it concerns characteristics of the environment outside the noun, it is outer epenthesis.

Inner epenthesis arises when the noun begins with a cluster of consonants beginning with s or x. An i is inserted before the noun. An example of a noun fulfilling these conditions is skola (school), for which the definite form is l-iskola.

Outer epenthesis occurs when the noun begins with a consonant and the article is not preceded by a word that ends in a vowel. An example of such a word is tak (he-gave-you). Hence tak il-ktieb/il-karta (he-gave-you the book/the paper). This is to be contrasted with tani l-ktieb/l-karta (you-gave-me the book/the paper). Note that outer epenthesis also occurs when the article is at the beginning of string. So when standing alone, we say il-karta.

Consonant assimilation takes place whenever the initial consonant of the noun is one the so-called “sun letters”: ċ, d, s, r, t, צ, x. In these circumstances, we write ċx-xemx (the sun), id-dar (the house) rather than il-xemx or il-dar.

2.2 Prepositions

Maltese prepositions that demonstrate interesting morphological behaviour¹ are shown below together with their nearest English equivalents:

¹ there are several others that do not
These forms are used when the preposition immediately precedes a noun without an article. This can be for a variety of reasons, e.g. (i) the noun is proper and doesn’t take an article (e.g. *lil Mike* - to Mike), (ii) the noun is indefinite (e.g. *bhal ghalliem* - like a teacher), (iii) the noun is definite in virtue of something other than the article ( *ta’ ommi* of my mother, where the possessive is formed with the suffix *i*).

There are some exceptions, however: *i* of *bi* and *fi* is replaced with apostrophe if the noun begins with a vowel, or with only one consonant; the result is joined to the next word as illustrated by *b’ommi* (with my mother), *f’Malta* (in Malta), *b’tifel* (with a boy).

The vowels of *ma’* and *ta’* are dropped before a word beginning with a vowel. Hence *m’ommi*, *t’Anna* (with my mother, of Anna).

### 2.3 Preposition + Article

When a preposition is immediately followed by the definite article, it is joined with it to form one word. An example is *sal-Belt* (to the city), which is formed by joining the preposition *sa* (to), the article and the noun. Notice that the result is still a single word.

An exception to this rule is the preposition *minn*, which, when joined with the article in this way, becomes *mil-*. However (see also below), before a noun beginning with *l*, one of the *ls* is dropped: *mil-Lebanon* (from Lebanon).

The prepositions *bi* (with) and *fi* (in) also follow this rule, e.g. *fil-forn* (in the oven), *bil-karozza* (with the car). However, if the noun begins with a vowel, the *i* is dropped, whether or not the vowel is the result of inner epenthesis. Hence we have *fl-iskola* (in the school).

Prepositions ending in *l* (*bhal, ghal, lil*) also have special behaviour. With nouns beginning with letters other than *l*, they behave normally, so we can have *ghall-Karnival* (for Carnival), *bhall-Ingliżi* (like the English). However, whenever the word after the article begins with the consonant *l*, that consonant is dropped (to avoid having three *ls* in succession. Hence we have *ghal-lukanda* (for the hotel), *lil-Libjan* (to the Libyan).

Consonant assimilation with the article takes place as before when the noun begins with a sun letter: so we have *fid-dar* not *fil-dar*.

The prepositions *ta’* (of) and *ma’* both include an apostrophe (which these cases stand in for an omitted silent letter *gh*) which is dropped when these prepositions are joined to the article: *tal-bieraħ* (of yesterday), *mat-tifel* (with the boy).
3 Computational Aspects

3.1 xfst

xfst is short for Xerox finite-state tool, and is one of a family of finite-state engines developed at Xerox for defining, exploring, and extending the potential of finite state technologies for language engineering purposes.

3.2 Description

In the description below we make heavy use of xfst’s replace operator (see [7]) used both conditionally and unconditionally.

**Character Classes and Words** We begin with the fundamental character classes

```plaintext
define V [a | e | i | o | u | ie];
define C [b | "_c" | d | f | ":g" | g | h | ":h" | j | k | l | m | n | m | p | q | r | s | t | v | w | x | "g_h" | ":z" | z];
```

together with the prepositions mentioned above:

```plaintext
define PREP [ {ma'} | {ta'} | {sa} | {bi} | {fi} | {minn} | {lil} | ["g_h" a l] | [b "_h" a l] | [ ":_g" o ] ];
```

A representative set of nouns is included, respectively beginning with a vowel, two consonants, a sun letter, and a consonant cluster starting with s.

```plaintext
define NOUN [ {ittra} | {ktieb} | {xemx} | {skola} ];
```

There are also two verbs respectively ending in a vowel and a consonant that will be used to demonstrate epenthesis.

```plaintext
define VERB [ {tak} | {tini} ];
```

3.3 Rules

**Conventions** “+” is used for morpheme/word boundaries, whilst “ws” stands for whitespace or beginning/end of input string (the latter defined as the regular expression

```
[+ | .#.]
```
Preposition without Article

\[
define \text{bificontract} \\
[ i \ %^+ \rightarrow ' || \text{ws} \ [b|f] \ _ \ [C \ V|V]];
\]

**Article** The article is represented on the lexical side by the abstract character L. We first present the rules that carry out inner and outer epenthesis:

\[
define \text{iepenthesis} \\
[[..] \rightarrow i || \ L \ %^+ \ _ \ [s \ C]];
\]
\[
define \text{oepenthesis} \\
[[..] \rightarrow i || \ C \ %^+ \ _ \ L];
\]

which are combined together by composition

\[
define \text{epenthesis} \\
oepenthesis .o. \ iepenthesis;
\]

The surface realisation of the article is governed by the following set of rules. The first converts the morpheme boundary into a “-”

\[
define \text{ajoin} \ [%^+ \rightarrow \ %- \ || \ L \ _];
\]

whilst the second ensures that the abstract L respects the behaviour demanded by Sun-letters.

\[
define \text{atran} \\
[ L \rightarrow "_c" \ || \ _ \ %- \ "_c"] \ .o. \\
[ L \rightarrow d \ || \ _ \ %- \ d] \ .o. \\
[ L \rightarrow s \ || \ _ \ %- \ s] \ .o. \\
[ L \rightarrow t \ || \ _ \ %- \ t] \ .o. \\
[ L \rightarrow x \ || \ _ \ %- \ x] \ .o. \\
[ L \rightarrow "_z" \ || \ _ \ %- \ "_z"] \ .o. \\
[ L \rightarrow 1];
\]

Finally the two rules are combined together, again using compostion. This time the order is relevant since the atran rules count on the transduction from “+” to “-”.

\[
define \text{art ajoin .o. atran};
\]

**Preposition with Article** The first rule states that if a preposition is followed by an article, they are assimilated, i.e. the morpheme boundary disappears. This is the basic preposition rule:

\[
define \text{prepbasic} \\
[ \ %^+ \rightarrow [..] || \text{PREP} \ _ \ L];
\]
The use of [...] rather than \(0\) facilitates using the rule in reverse.

Next, the exceptions. For \(bi\) and \(fi\) the \(i\) disappears if the noun ahead of the article starts with a vowel.

\[
\text{define bfil2bfl} \\
[ i \rightarrow [..] || ws [b | f] \_ L \%+ V ];
\]

The ordering of this rule is delicate, since to operate properly it has to be invoked take place after (inner) epenthesis, which could potentially insert a vowel after the article.

A second exception is the behaviour shown by prepositions that end in \(l\). As already mentioned, this can be assimilated under a general rule that prevents more than three identical consonants from ever appearing adjacent to each other. Unfortunately it is difficult to state a rule of this degree of generality in xfst: we are forced to address the specific case specific case with a rule like this - and the ordering is still critical.

\[
\text{define l32} \\
[ l L \rightarrow L || _ \%+ l ];
\]

3.4 Results

The problem under investigation can be regarded thus: given our definitions, we want to define a system that will transduce between strings generated by the expression

\((\text{VERB} \%+) \text{PREP} (\%+ L) \%+ \text{NOUN}\)

and the underlying lexical representation.

4 Conclusion and Future Work

This article is an incomplete draft of a forthcoming paper that will include examples from an online demonstration, a discussion of the quality of results, and suggestions for extending the analysis to other linguistic phenomena in Maltese. The accompanying talk will address some of these points.

5 Acknowledgements

I should like to thank the University of Malta for supporting the research described in this draft. I also thank my colleagues at in the Maltilex project and particularly Ray Fabri, for their helpful comments. Finally, I am indebted to Ken Beesley for guiding me through some of the less intuitive features of xfst.
References


A Risk Driven State Merging Algorithm for Learning DFAs

Sandro Spina
Department of Computer Science and AI, University of Malta

Abstract. When humans efficiently infer complex functions from a relatively few but well-chosen examples, something beyond exhaustive search must probably be at work. Different heuristics are often made use of during this learning process in order to efficiently infer target functions. Our current research focuses on different heuristics through which regular grammars can be efficiently inferred from a minimal amount of examples. A brief introduction to the theory of grammatical inference is given, followed by a brief discussion of the current state of the art in automata learning and methods currently under development which we believe can improve automata learning when using sparse data.

1 Grammatical Inference

A typical definition for learning would be the act, process, or experience of gaining knowledge. Within the field of machine learning this process of gaining knowledge is achieved by applying a number of techniques, mainly those relying on heuristic search algorithms, rule-based systems, neural networks and genetic algorithms. This short report focuses on the learning of regular grammars (those languages accepted by finite state machines) by making use of heuristic search algorithms to direct the search. The process of learning grammars from a given set of data is referred to as grammatical inference (GI).

Automata learning is the process of generalizing from a finite set of labelled examples, the language (FSA) which generated them. Let us say that we’ve got the +ve example set \{10, 20, 30, 80\}. Positive since these examples are labelled “accepted” by the target language. We can immediately infer that the target language is that of integers divisible by 10 (or rather strings whose length is divisible by 10). However, by overgeneralizing we can also infer that the language is that of even integers (strings whose length is divisible by 2). Both are correct; however as we’ll be outlining in the next section, this example illustrates how vital the training sample is (both +ve and -ve samples), for efficient, correct grammatical inference.

The field of grammatical inference finds practical applications within areas such as syntactic pattern recognition, adaptive intelligent agents, computational biology, natural language acquisition and knowledge discovery as illustrated in [6].

In the next section we will be discussing some theoretical background.

2 Preliminaries

Automata learning or identification can be formally expressed as a decision problem.
Given an integer \( n \) and two disjoint sets of words \( D_+ \) and \( D_- \) over a finite alphabet \( \Sigma \), does there exist a DFA consistent with \( D_+ \) and \( D_- \) and having a number of states less than or equal to \( n \)?

The most classical and frequently used paradigm for language learning is that proposed by Gold [3], namely language identification in the limit. There are two main variations of this paradigm.

In the first one the learner can make use of as much data as necessary. The learning algorithm is supplied with a growing sequence of examples compatible with the target automata. At each step the learner proposes a hypothesis DFA, representing the guessed solution. The algorithm is said to have the the identification in the limit property if the hypothesis (consistent with all learning data) remains unchanged for a finite number of guesses. In the second case the number of available learning examples is fixed and the learning algorithm must propose one hypothesis from this set of examples. This algorithm is said to have the identification in the limit property if, for any target machine \( A \), it is possible to define a set \( D'_A \) of training examples called the representative sample (characteristic set) of \( L(A) \) [4]. Our work currently focuses on this second variation, were we’re currently focusing on determining any lower bounds for the sparsity of the training data in order to be able to identify certain classes of regular languages.

Gold [3] has proved that this decision problem is NP-complete, however if the sets \( D_+ \) and \( D_- \) are somehow representative of the target automaton, there exist a number of algorithms that solve the considered problem in deterministic polynomial time.

In the next section we’ll be describing two main GI algorithms.

## 3 Learning Algorithms

The first algorithm is due to Trakhtenbrot and Barzdin [5]. A uniformly complete data set is required for their algorithm to find the smallest DFA that recognizes the language. Their algorithm was rediscovered by Gold in 1978 and applied to the grammatical inference problem, however in this case uniformly complete samples are not required. A second algorithm, RPNI (Regular Positive and Negative Inference) was proposed by Oncina and Garcia in 1992. Lang [5] proposed another algorithm that behaves exactly in the same way as RPNI during the same year. The RPNI algorithm is based on merging states in the prefix tree acceptor of the sample. Both algorithms are based on searching for equivalent states. These algorithms had a major impact in the field, since now languages of infinite size became learnable. Lang proved empirically that the average case is tractable.

Different control strategies (heuristics) can be adopted to explore the space of DFA constructions. At each step a number of possible merges are possible, thus the merging order of equivalent states determines the correctness of the generated target language hypothesis. To make things clear let us consider the regular expression \( ab^*a \), with \( D_+ = \{aba, aa, abbba\} \) and \( D_- = \{b, ab, abbb\} \). The Augmented Prefix Tree Acceptor (APTA) for these training sets is shown in figure 1.

![Figure 1: APTA](image)

Note that final (accepting) states are labelled 1, non-final (rejecting) states are labelled 0 and unknown states are marked ?. The task of the learning algorithms is to determine the correct labelling for the states marked with a ?. The learning algorithm proceeds by merging states in the APTA, until no more merges are possible.

Rodney Price [2] proposed an evidence driven heuristic for merging states. Essentially this algorithm (EDSM) works as follows:

1. Evaluate all possible pairings of nodes within the APTA

\[\text{Evaluate all possible pairings of nodes within the APTA}\]
2. Merge the pair of nodes which has the highest calculated evidence score (pair of nodes whose subtrees share the most similar labels).
3. Repeat the steps above until no other nodes within the APTA can be merged.

Figure 2 shows the process of merging states for the APTA shown in figure 1 using the EDSM program available at the Abbadingo web site. The resulting automaton illustrates clearly some of the shortcomings of the EDSM algorithm. Figure 3 illustrates the search space for the learning algorithm. The target automaton lies somewhere between the APTA that maps directly the training data and the universal acceptor.

Our current research is devoted at improving these search heuristics. The difficulty of detecting bad merge choices increases as the density of the training data decreases, because the number of labelled nodes decreases within the APTA. In the algorithm we are proposing a risk value is associated with each potential merge. During the initial phases of this project we are using various data structures (such as suffix trees) and string algorithms that are helping the algorithm in determining the risk.
factor for each merge. The algorithm selects the pair of states with the lowest merge risk value and proceeds. Our primary aim is to implement a DFA learning algorithm with variable heuristic parameters that is able to learn target languages from low density sparse training examples.

References

Regulating Electronic Commerce using XML
Development of standards for the regulation and widespread interfacing of e-commerce

Kenneth Spiteri
Department of Computer Science and AI,
University of Malta

E-commerce is much more than the opening up of a new, online sales channel for an enterprise, it is about using technology to streamline a company’s business model, creating savings and increasing efficiency, lowering costs and establishing closer, more responsive relationships with customers, suppliers and partners. A lot of businesses are trying to make further use of their presence online, with increased levels of interaction and services, mapping existing physical business processes in a virtual environment. The result of this is an increase of financial transactions being carried out in what has become the commercial frontier of cyberspace. Whilst the existing problems of carrying out business remain, a myriad of new problems have been created due to the different implementation strategies undertaken by these enterprises and the many technologies available to carry this out. The resulting situation is that of separate commercial “islands” created that whilst functioning adequately in their specific areas cannot extend in scope, thus defying the ontology of commerce. The need to standardise on these considerations becomes more apparent. In this regard, attempts to regulate the conduct of financial transactions and its relevant aspects on the Internet medium, are already underway by various bodies with varying degrees of success. The aim is that of protecting the rights of consumers and users whilst building public confidence in e-commerce. The establishment of new technologies and methods of conducting electronic business are constantly posing new problems that have to be considered, thus making this process even more difficult.

This research analyses current e-commerce standards and technologies, and their validity in safeguarding online business, identifying where they are lacking and in what they might fail if not regulated by operational parameters. The solution aims at providing a better insight on how to mirror this physical regulation of the media with the virtual regulation of the processes constantly occurring online, defining parameters by which transactions between heterogeneous systems should abide, thus enforcing the privacy and security of the transactions and their mandates. The amalgamation of the various facets of the research carried out would enable the implementation of a possible set of such parameters, within the development of a prototype standard for the virtual regulation of online business.

While various entities have developed their own standards as part of a solution to this problem, these standards vary in solution focus, use and requirements; a number of which try to solve specific issues such as financial reporting and EDI integration, for example, rather than providing a level concrete solution. To substantiate this point, some of the specific Standard implementations can in fact plug in together to provide a valid solution in themselves, as these complement each other through their separate yet related areas. Standards in themselves should not be in direct competition with each other, as this would leave the current situation in a status quo position. Rather, they prove their validity for the enterprises adopting them, due to the requirements of the individual situations, but with the facility to interact seamlessly with other standards.

From this research, a framework for a new general business standard named ECXML (Electronic commerce extensible mark up language) has been set up and implemented, allowing an enterprise to use this standard to provide a description of the business structure, irrespective of the area of business, to allow for the integration whilst the working system being developed aims to demonstrate the integration that can be achieved though the use this standard and other standards, that enable the integration of such systems without effecting, as much as possible, the particular enter-
prise’s online business processes, while still being flexible enough to allow for the future evolution of the standard to cater for products and services not envisaged at the point of creation.

References

Achieving User Adaptivity in Hyperspace with HyperContext

Christopher Staff

Department of Computer Science and AI, University of Malta

Abstract. HyperContext is a framework for adaptive and adaptable hypertext. In any hyperspace, each piece of information (e.g., contained in a document or node) is normally juxtaposed by other information via links. Two or more hypertext users may encounter the same document although they may have followed different paths to reach it. Those two users may well describe different aspects of the document as relevant to their needs and requirements. The HyperContext framework allows users to create different interpretations of information in context, which will also be available to future users. (Work adapted from [6])

1 Background

Adaptive Hypertext Systems (AHSs) [1] fall into two very general categories. We distinguish between domain-specific adaptive hypertext systems, such as AHS-based Intelligent Tutoring Systems (ITSs), and general-purpose AHSs. We cannot, without incurring great expense, as accurately model the user and the domain in general-purpose AHSs as can be achieved in narrow domain-specific ITSs. In HyperContext, a framework for general-purpose AHSs, we monitor the scope of a user’s short-term interest through a construct named the context session. As long as the user is traversing through hyperspace within the same context session, and has not hyper-leaped out of it, we assume that she is still searching for the same information, and that she has not yet located it.

In Information Retrieval (IR) systems, a document is frequently represented by a single vector of weighted terms, which may then be used to attempt to satisfy a large number of different requirements (see, for example, [4]). Although users are sometimes able to give relevance feedback [5], this information is normally used to modify the user’s query, rather than to modify the document’s vector representation. As a result, an IR system used by another user with an identical information need normally cannot take advantage of its previous users’ experience to improve its quality of service, whereas a HyperContext hypertext can as its users record how the information they access is relevant to them.

2 HyperContext Overview

Interpretations permit a hyperspace to be partitioned as it is traversed. A context in HyperContext is an arbitrary container in which data can be interpreted to acquire meaning. We minimally

---

1 We use context in a similar way to McCarthy to mean an environment that enables data to be attributed with meaning [3].
associate context with a document and a link source anchor in the document. A document’s out-links are associated with interpretations, so the same document in different contexts can have different out-links, or common out-links which have different destinations in each interpretation. This gives us the notion of a context path, a sequence of interpretations linked in some context. As a user browses through hyperspace, a node can be contextually relevant if it is on the same context path as the user and relevant to a user’s interests. Other documents in the hypertext which are relevant to a user are superficially relevant. While a user browses through a hyperspace, the interpreted documents accessed on the path of traversal form the user’s context session. A user browsing through hyperspace may make superficially relevant information contextually relevant by extending a link to it from any node in the context session.

Users searching for information are supported by three Information Retrieval mechanisms: Traditional Information Retrieval (TIR), Information Retrieval-in-Context (IRC), and Adaptive Information Discovery (AID). TIR enables a user to specify a query and presents the user with all relevant interpretations, regardless of context. IRC presents contextually relevant interpretations in response to a user supplied query. AID utilises a short-term user model to assimilate a user’s short-term interest, based on the context session, and can automatically generate a query on behalf of the user. Superficially relevant information is recommended by AID to the user via “See Also” links. If, following a search, a user hyperleaps to a node containing superficially relevant information she is given the option to make it contextually relevant by extending a link to it from within the current context session, otherwise a new context session is initiated. On the other hand, HyperContext can guide the user to contextually relevant information by recommending links through the hyperspace.

The short-term user model is initialised at the beginning of a new context session. We distinguish between a long-term and a short-term interest. A long-term interest is one which persists across many context sessions, and perhaps lasts for weeks, months, or even years. A short-term interest is transient. It may extend over a small number of context sessions, but it is unusual for it to last for long, although short-term interests can develop into long-term interests. We express the user’s short-term interest as a function of the interpretation of documents that the user has seen in the current context session. The user’s perceived interest in the current document is represented as a salient interpretation. Salient interpretations are combined in the user model according to a weighted scale of confidence in the salient interpretation’s usefulness in identifying a relevant document.

3 HyperContext Supports Adaptable Hypertext

The HyperContext framework permits a hyperspace to be adapted by its community of users to reflect how the information is actually consumed. Users are able to reuse information, regardless of who authored or “owns” the information, by creating new links between existing nodes. Users are also able to describe the information in the destination node which is relevant to them, to provide an interpretation of the information in the node. Each interpretation of a node is represented by a vector of weighted terms. The parent node containing the link source anchor and the link itself provide the context in which the destination node will be interpreted whenever it is accessed via that link. The interpretations of a node collectively reflect the different ways of describing the information contained in the node.

The interpretations of information are searchable and retrievable through an interface between the HyperContext framework and an external information retrieval system. The HyperContext prototype interfaces with SWISH-E [2], which provides external information indexing and retrieval services to HyperContext. When a user searches for information, HyperContext invokes the external
IR system and retrieves interpretations of documents which are relevant to the query. Depending on which search mechanism the user invoked, HyperContext will either present the user with a ranked list of relevant interpretations (TIR), or it will guide the user to a contextually relevant interpretation by recommending links to follow along a context path (IRC). Non-adaptive hypertexts normally cannot guide users to information without the hypertext author having first created a purpose-built trail. On the other hand, adaptive hypertext systems can guide users to relevant information using trails or paths of traversal frequently travelled by previous users. However, in HyperContext we distinguish between contextual relevance and superficial relevance to guide users to relevant information along a context path which other users have previously created.

4 Adapting to the user

A benefit of adaptive hypertext systems is that they are able to automatically or semi-automatically determine a user’s interests [1]. In HyperContext we distinguish between a user’s short-term interest and her long-term interest. We assume that a user is likely to require greater support in her search for information to satisfy a short-term interest, because she is likely to be unable to accurately represent her information need. We must also detect when the topic of a user’s short-term interest has changed, otherwise our representation of the user’s interest may be contaminated by no longer relevant information.

We construct a model of the user’s short-term interest based on the interpretations of nodes that she has accessed in the context session, assuming that each accessed interpretation in the context session is only partially relevant to her information need (otherwise she would have located the required information and terminated the context session). As an accessed interpretation is considered only partially relevant to the information need, we establish which terms are likely to be relevant and the degree to which they are relevant by deriving a salient interpretation of the node. The salient interpretation is derived using a modification to the Rocchio relevance feedback method, which compares the accessed interpretation of a node to all the other interpretations of the same node to establish which terms are likely to best represent the user’s interest in the node. A scale of confidence is used to weight each salient interpretation of the interpretations accessed during the context session, to reflect HyperContext’s confidence in each salient interpretation’s ability to contribute information about the user’s short-term interest. The weighted salient interpretations are finally combined as a model of the user’s short-term interest.

The Adaptive Information Discovery (AID) search mechanism is an autonomous tool which, if active, generates a search query on the user’s behalf by extracting terms from the short-term user model. The user can be guided to information that is contextually relevant as well as being presented with a list of superficially relevant “See Also” references.

5 Evaluation and Results

HyperContext was evaluated in 2000 [6]. We automatically selected a series of documents in a number of context paths and then, for each context path, used Adaptive Information Discovery to search for a relevant document. Additionally, a non-adaptive technique was used to create a query to search for a control document for each path. Users were asked to read the documents in an entire context path and then provide relevant judgements for each of the AID-selected and control documents. The AID-selected document was given a higher or equal relevance judgement a statistically significant number of times. The experiments will be repeated in 2003 on a re-constructed implementation of HyperContext.
6 Conclusion

The HyperContext framework is the result of research which crosses the boundaries of the domains of adaptive hypertext, hypertext, user modelling, information retrieval and context. We believe we have contributed to the area of adaptive hypertext by incorporating automatic relevance feedback mechanisms into the derivation of the model of the user’s short-term interest. We also believe we have extended research into adaptive hypertext systems by incorporating explicit representations of context into hypertext systems which permits multiple interpretations of the same information to be represented and manipulated to give individual users adaptive navigation support. These conclusions are supported by the experimental results obtained from an implementation of the important adaptive features of the HyperContext framework.

References

Learning About the Learner: 
Deducing Learner Characteristics From Their Interaction 
with e-Learning Tools 

Ingrid Talbot 
Faculty of Education, 
University of Malta

1 Introduction

The study seeks to explore how e-learning is tackled by learners and thus investigate how hypermedia tools help to identify specific learner characteristics. Such characteristics can be detected by means of assessing the manner in which adult learners perform in a number of given e-learning tools. The different navigating and learning strategies of individual learners will be observed. In this way a knowledge acquisition strategic pattern can be established.

Unlike other traditional teaching and learning tools, where much of this information tends to be lost within the pedagogical process, e-learning can thus be seen under the light of its potential to acquire knowledge of every learner on an individual basis. Such an asset gives the possibility to collect information about the learner not only in the traditional manner of assessments, but also through:

1. through the way the users interacts with the e-learning tool facilities; 
2. through the order in which the user navigated through the material presented by the tool.

Hence, the uniqueness of hypermedia lies in the fact that it recognises that people learn in different but interwoven ways.

2 Related Work

The research area in question mainly involves the effects of cognitive styles on hypermedia navigation. In spite of the fact that there has been an increased use of hypermedia to deliver teaching and learning material, there still remains much to be explored and learnt about how different learners perceive such systems.

Most research in this area deal with non-linear and linear learning, field-dependent and field-independent learners, learner control, navigation in hyperspace and learning effectiveness. Chen and Macredie mention the uniqueness of hypermedia in that:

*there is no one linear path through the program but a multitude of branches in which a learner can explore a subject matter at his/her own pace.* [3]

Ayersman and Minden reinforce this concept by suggesting that *hypermedia has the capacity to accommodate individual learning style differences.* [1]
Messick in turn speaks of the importance of acknowledging cognitive styles as a unique characteristic of each and every learner. And it is for this reason that the development of hypermedia-based learning has an important role in terms of these learning patterns which are indications of:

*the users processing habits, the users typical modes of perceiving, thinking, remembering and problem solving.* (Messick in [3])

Ford and Chen mention that among the various dimensions in the research area concerning cognitive styles, Field-Dependence and Field-Independence has emerged as one of the most widely studied having a wide applicability to education.

The following outlines the differences between Field-Dependent and Field-Independent Learners

**Field-Dependant learners:**

- They find it difficult to restructure new information and forge links with prior knowledge
- Their personalities show a greater social orientation
- They experience surroundings in a relatively global fashion, passively conforming to the influence of the prevailing field or context
- They demonstrate fewer proportional reasoning skills
- They prefer working in groups - They struggle with individual elements
- They are externally directed
- They are influenced by salient features
- They accept ideas as presented

**Field-Independent learners:**

- They are able to reorganize information to provide a context for prior knowledge
- They are influenced by social reinforcement
- They experience surroundings analytically, with objects experienced as being discrete from their backgrounds
- They demonstrate greater proportional reasoning skills
- They prefer working alone
- They are good with problems that require taking elements out of their whole context
- They are internally directed - They are individualistic
- They accept ideas strengthened through analysis

Adapted from [24, 16, 19]

Hypermedia tends to present learning in a non-linear format. This implies that Field-Independent students are relatively capable of setting the learning paths by themselves in hypermedia programs with non linear presentation. On the other hand, field-Dependent students seem to prefer to have a fixed path to follow in linear learning programs.

Hence a new challenge is raised in view of the fact that adult learning, particularly technology-enhanced learning, is still in its development stages. Delving into new content areas themselves, in most circumstances is the way to acquire more knowledge in this field. As we grow in our understanding about what it takes to teach adults effectively, we are seeing distinct patterns in how adults tend to learn. Although it is difficult to outline all of these learning styles since learning is inherently a personal quest, and therefore, individuals may have diverse learning patterns, one can still attempt to identify some of the most common and critical patterns to know prior to developing adult learning opportunities.

In discovering these patterns, one can identify learning strategies referring to methods that adult students use to learn by means of accommodating the differences in personal and situational characteristics of every student. Thus,
1. Adult learning programs should capitalise on the experience of participants.
2. Adult learning programs should adapt to the aging limitations of the participants.
3. Adults should be challenged to move to increasingly advanced stages of personal development.
4. Adults should have as much choice as possible in the availability and organisation of learning programs.

Ultimately learning can and should be a lifelong process. It should not be defined by what happened early in life, only at school since learners constantly make sense of their experiences and consistently search for meaning. In essence, learners continue to learn. For this reason, the age factor plays a very important role and in this regard, it is necessary to make a clear distinction between young and adult learners and thus the implication of maturity on the notion of lifelong learning. Age will be considered in the light of the way it influences learning and the style adopted for knowledge acquisition. Thanks to individual learning processes, learning environments and learners themselves can be re-created. Adult education can truly act as a catalyst for change in peoples lives, strengthening self-confidence and sparking higher goals in learning, work, and life.

3 The Proposed Project

3.1 Overview

Various traits of the user can be deduced through the pattern he or she uses whilst navigating and getting hands-on experience of the subject content contained within an e-learning tool. Through the analysis of the use of these tools, information about the user can be deduced and such observations will be subsequently used to improve the learning experience for the user.

The background to the following dissertation will be an exploration of three main branches namely:

1. learning styles
2. age learning issues and
3. experience with a particular electronic learning environment

All of the above three elements, although in an independent manner, will help to create a better picture, thus giving a valid evaluation of the user. In turn, this will facilitate the adoption of the right teaching and learning methodology.

This three-fold approach will thus focus on the main characteristics of the adult learner. It is interesting to note that the above-mentioned are independent variables. This implies that even though all of them will help create a clearer learners profile, they are not mutually dependent. The learners experience, age and learning style all provide feedback which will serve as input to the observational and analytical stage of the project.

3.2 Analysis

The analytical phase of the project will have two main facets. These two distinct components will be that of:

1. Discovering which learning strategies the user makes use of in order to gather the necessary information from the e-learning tool, throughout the navigation process.
2. Determining to what extent has the learning of the actual content within the e-learning tool has been effective

The term tool familiarity refers to the learners experience to find his or her way through in a particular e-learning environment. One cannot but recognise that the user interface may be quite complex. Thus, for the sake of the feature which is being explored, it is important not to confuse learning navigation with the tool with the actual subject being taught.

Through the exploration of adults of different age groups, the age factor of adult learners will be explored together with its relevance on the learning pattern of the individual.

The effectiveness of e-learning tools will be measured through a form of assessment. In comparison to traditional pedagogy, the usefulness of such an approach is that it can be varied in such a way so as to accommodate the learning needs and specifications of the individual learner.

An additional advantage of e-learning is that it puts the learner as the focal point of the whole process. The student is given the freedom to choose from a variety of patterns through which knowledge can be acquired and mastered. A field to be explored will be in fact that of verifying whether a particular age can be identified with particular learning patterns or whether these learning styles vary unconnectedly. A connected issue will be that of seeing to what extent one could possibly deduce the learning style of a student simply by observing the pattern used when navigating the information which is presented within the learning tool.

The great potential of hypermedia is that ultimately it provides non-linear learning and gives the user legroom for experiential learning. Another aspect to be considered is the manner in which information is presented and made accessible within these media, thus it enhances constant user interaction. An additional strength attributed to hypermedia is that it can be reconfigured according to the style adopted during the navigation. All this takes place exclusive of the users awareness of the ongoing process. All this happens on the fly since the presence of a machine in the background coordinates this activity. However, the effectiveness of hypermedia still has vast areas which are still to be explored.

3.3 Deductions: Aims of the Project

Since different learners profiles would have been created subsequent to the analysis stage of the project, the following proposals will be put forward as part of the study:

1. Learning tools should be built whilst keeping in mind the age factor of the typical learner since this is highly influential on the way people learn and will in turn impinge on the presentation style of prospective such tools.
2. Learning tools can be reconfigured dynamically since as the learner interacts with the user interface, the machine which lies in the background learns about the learner and shapes the different knowledge presentation components of the learning tool.
3. E-learning tools provide easy access to information about how the learner navigates through the new information. By analysing the users logs, one can obtain useful information about the knowledge content being explored and the areas which the user finds particularly difficult. This in turn will act as an aid to syllabi reformation so as to boost learning efficiency.
3.4 Practical Applications of Results

Access has been given to a number of commercial e-learning tools whereby the researchs findings can be applied.

1. Avicenna is a UNESCO led European Commission funded project aiming to accelerate the adoption of Open Distance Learning in the Euro-Mediterranean region with partners come from fifteen Mediterranean countries. The goal of the project is to create a virtual campus, through the training of people from the participating institutions and funding of development of e-learning units by these institutions.

2. A company which is currently involved in various project within the local context is Seasus New Media whose services includes integrated web solutions and distance learning. The company will give access to its tools, including those still in the pipeline, for the sake of analysis in the study.

4 Conclusion

This research area clearly indicates the significance of lifelong learning in our day. This term denotes the learners participation in any type of education or training course. This phenomenon encapsulates initial education, further education, continuing and further training, training within the company, apprenticeship, on-the-job training, seminars, evening classes and distance learning.

In this day and age Lifelong Learning is a very important issue since one cannot but keep up-to-date with the times. Malta has been described in light of its strong potential for Electronic Learning which is the key to future learning due to its increasing significance. The global diffusion of e-learning can be surely put to the avail of adult education.

The ultimate aim remains that of empowering the learning community and placing the learner at the very heart of his or her local commune. The ideal would be to map and track existing lifelong learning provisions, internally and externally; to analyse the current position, looking for strengths, weaknesses, gaps and overlaps thus creating a more joined-up approach to lifelong learning, with a constant focus upon the needs of end users, not the providers.

References


A Summary of Research in System Software and Concurrency at the University of Malta: Multithreading

Kevin Vella
Department of Computer Science and AI, University of Malta

1 Introduction

Multithreading has emerged as a leading paradigm for the development of applications with demanding performance requirements. This can be attributed to the benefits that are reaped through the overlapping of I/O with computation and the added bonus of speedup when multiprocessors are employed. However, the use of multithreading brings with it new challenges. Cache utilisation is often very poor in multithreaded applications, due to the loss of data access locality incurred by frequent context switching. This problem is compounded on shared memory multiprocessors when dynamic load balancing is introduced, as thread migration also disrupts cache content. Moreover, contention for shared data within a thread scheduler for shared memory multiprocessors has an adverse effect on efficiency when handling fine grained threads.

Over the past few years, the System Software Research Group at the University of Malta has conducted research into the effective design of user-level thread schedulers, identifying several weaknesses in conventional designs and subsequently proposing a radical overhaul of the status quo to overcome these deficiencies. Various results have been published in academic conferences and journals [1–4]; this brief report highlights the principal findings. The related problem of communication and I/O bottlenecks in multithreaded systems and contemporary computer systems in general is discussed elsewhere in these proceedings [5].

2 The Old Testament

Many multithreaded environments utilise kernel-level threads as a lightweight alternative to heavy-weight operating system processes. Kernel-level threads share a single address space to enable sharing of data and to minimise thread creation and context switching times. Kernel entry is still required when threads are created, whenever they communicate or synchronise, and at every context switch. Furthermore, the threading model is implemented within the kernel and is therefore dictated by the kernel. Systems such as Microsoft .NET make direct use of kernel threads as a vehicle for driving multithreaded applications.

The expense and inflexibility of kernel involvement can be eliminated through the use of user-level threads, which operate entirely at the user level. Thread operation times as low as a few tens of nanoseconds can be achieved using user-level threads. Since the kernel is oblivious to user-level threads, a two-level thread hierarchy, with several user-level threads riding over a small pool of kernel-level threads, is employed to gain access to multiple processors and other kernel resources. This model, which is increasing in popularity, is utilised in this research, as it provides sufficient flexibility for experimentation with alternative scheduling strategies and yields by far the fastest implementations.

In this section we mention efforts at implementing uniprocessor and multiprocessor user-level thread schedulers based around the traditional single shared run queue approach.


2.1 Uniprocessor Thread Scheduling

Most uniprocessor user-level thread schedulers in existence utilise a single FIFO queue to hold descriptors for runnable threads. Cooperative round robin scheduling is used, with threads being descheduled whenever inter-thread communication or synchronisation occurs, or when an explicit ‘yield’ operation is invoked (of course, such descheduling points may be inserted automatically by a preprocessor). Thread operation times can be as low as 20 nanoseconds, particularly when the compiler passes register usage information to the scheduler to restrict thread state saving overhead. Examples of this genre include our Smash [6], Kent’s KRoC [7] and CERN’s MESH [8].

The problem with such a naive scheduling strategy is that frequent context switching (a characteristic of fine grained threads) disrupts the operation of the locality principle, on which cache hits depend. The expense of repopulating the processor’s cache with a newly dispatched thread’s footprint becomes significant when viewed in relation to the shortened thread dispatch time. To make matters worse, an individual thread is unlikely to accumulate a significant cache footprint by itself: only when threads are considered in groups can a long term cache footprint be identified.

2.2 Multiprocessor Thread Scheduling

The obvious way of extending a scheduler to operate on a shared memory multiprocessor is to wrap a monolithic lock around the scheduler’s entry and exit points to protect against corruption arising from concurrent access to the scheduler’s internal data structures. Through further refinement, the sizes of the critical sections may be reduced, and independent internal structures may be guarded by separate, fine-tuned locks to reduce contention. It emerges that the shared run queue is the data structure that will experience most contention as the multiple processors compete to fetch the next thread to execute. The locking method used must exhibit low latency, thus excluding the use of kernel-assisted locks such as semaphores. Usually a spin-lock variant designed to minimise memory bus traffic is used. SMP-KRoC [2] was implemented in 1998 following this design. Subsequently, work undertaken at the University of Malta adapted CERN’s MESH [8] in a similar fashion to produce SMP-MESH [9]. A variant of our scheduler, Shared-Smash [6], also follows this design. We daresay that most multiprocessor user-level thread schedulers that balance thread load across processors operate on the same lines.

While schedulers adopting this approach exhibit perfect balancing of thread load across processors, the shared run queue causes threads to be indiscriminately migrated to processors on which they would not have recently executed. As a consequence, newly dispatched threads are unlikely to find any of their data footprint in the local cache. Moreover, when fine grained threads are being executed and the thread dispatch frequency is high, contention amongst the processors for access to the shared run queue inevitably spirals, despite efforts to minimise the critical section size.

3 The New Testament

Having glanced at what may be considered to be ‘traditional’ thread scheduler design and identified the following shortcomings:

– poor cache utilisation on uniprocessors;
– worse cache utilisation on shared memory multiprocessors; and
– high levels of contention for the run queue on shared memory multiprocessors;

we now provide a brief overview of our proposed solutions.
3.1 Thread Batching

Thread batching, first proposed in [10], involves grouping threads together into coarser grain entities termed batches. In batch schedulers, the scheduled entity is a batch of threads rather than an individual thread. A processor obtains one batch at a time from a batch pool and services the threads on the batch for a fixed number of thread dispatches, before depositing the batch back on the batch pool and acquiring the next batch. If the threads within a batch are scheduled repeatedly within the same batch dispatch and the combined memory footprint of the threads in the batch fits within the processor’s cache, cache exploitation will naturally improve. In practice, this approach regains some of the data access locality which multithreading disrupts in the first place, even on unprocessors.

On shared memory multiprocessors, batching algorithms yield even better results. Since the dispatch time of an entire batch is large when compared to the dispatch time of an individual thread, sufficient cache reuse is carried out within a single batch dispatch to dwarf the cache-related expense of thread or batch migration across processors. Moreover, if processors fetch batches from a shared run queue, the relatively large batch dispatch time reduces the frequency of access to this shared data structure, thus reducing contention for it while maintaining a balanced batch workload.

Batching can also be used to decrease the incidence of false sharing, since threads accessing data in a common cache line may be batched together. Moreover, when balancing load across processors, migrating threads in batches (whether in a shared batch run queue environment or otherwise) reduces the contention that arises when migrating multitudes of individual threads.

Batching gently nudges the memory access patterns of multithreaded applications to fit into a more sensible regime. It also coarsens the granularity of real concurrency in a manner which is dynamic and totally transparent to the developer.

Uniprocessor thread batching was first implemented in an experimental version of uniprocessor KRoC [10], with encouraging results: performance of fine grained multithreaded applications was improved by as much as 28%. As the gap between processor and memory speeds grew, this figure was only bound to improve. In fact, two years later, Unibatch-Smash registered improvements of up to 100%. On shared memory multiprocessors, additional benefits were made apparent using SmpBatch-Smash, a shared run queue multiprocessor batch scheduler. Various other scheduling arrangements utilising batching were implemented with positive results. Detailed accounts of the inner workings and performance of the thread batch schedulers implemented at the University of Malta may be found in [6, 4, 1].

3.2 Appling Lock-free Data Structures and Algorithms

Traditionally, access control to concurrent data structures relies on the use of locks. An oft-overlooked alternative method of synchronisation is available through the considered use of lock-free structures and algorithms, which dispense with the serialisation of concurrent tasks. Lock-free data structures rely on powerful hardware atomic primitives and careful ordering of instructions to protect them from unsafe concurrent access.

Valois [11] discusses lock-free techniques in detail and supplies various definitions of relevance. Lock-free data structures may have a further two properties: they may be non-blocking and wait free. A lock-free data structure is termed non-blocking if some operation on it is guaranteed to complete in finite time. When used for thread scheduling and inter-thread synchronisation, non-blocking data structures have other advantages, including stronger fault tolerance, deadlock freedom, removal of the extended spinning problem on multiprogrammed multiprocessors, and in priority-based schedulers, elimination of priority inversion within the scheduler routines. Unfortunately,
the non-blocking algorithms usually rely on retries to recover from unexpected alterations performed concurrently by other processors. This can result in unpredictable delays and starvation under high contention. Furthermore, the use of COMPARE-AND-SWAP in most of the algorithms brings about the ABA problem [11], which necessitates complex support for memory management to avoid it.

If every operation on the data structure is guaranteed to complete in a fixed number of operations the structure is said to be wait-free. Wait-free data structures, as discussed by Herlihy [12], do not suffer from the ABA problem and do not ever require retries. As a consequence, starvation is eliminated and the maximum number of instructions executed in the algorithms is fixed at compile-time.

Lock-free algorithms and data structures are well documented in academia, but an account of the application of this technique in a scheduler is hard to come by. We were unable to locate publications describing a wait-free scheduler implementation other than our own effort. The use of wait-free techniques within Wf-Smash [3, 1], a multiprocessor thread scheduler designed and implemented locally, removes all of the locks and critical sections that protect internal data structures. This scheduler also utilises thread batches to coarsen the granularity of thread migration. As a result, high levels of efficiency are sustained at very fine thread granularities on shared memory multiprocessors, where all other thread scheduling strategies break down due to contention and cache misses.

4 Conclusion and Further Work

We have conducted an investigation into the effectiveness of cache-conscious scheduling using batches, and the exclusive use of wait-free synchronisation techniques in a thread scheduler for shared memory multiprocessors. The experimental results obtained indicate that a significant reduction in execution times can be gained at fine levels of thread granularity through the use of such techniques, both individually and in combination.

The implications of wait-free thread scheduling are numerous. Any path through the scheduler involves the execution of a fixed number of instructions, thus enforcing an upperbound on the time spent in it. In multiprogrammed systems, the extended spinning that typically ensues when a kernel thread is descheduled whilst holding a lock is no longer an issue. In general, contention for shared structures within the scheduler is reduced considerably. We believe that a substantial performance improvement can be achieved on shared memory multiprocessors by avoiding any form of locking in such scenarios, including blocking locks as well as spin locks. This approach should be at least considered as a practical alternative to the status quo.

Further experiments with real applications are required to gather information about the performance of batching under more realistic conditions. It should be noted that batch-based thread scheduling as presented here may be subject to problems when the pre-set batch size limit is greater than the total number of threads being executed in the application, since all threads would be serviced by a single processor. While this can be advantageous in identifying situations where parallel processing is not worth the effort, pathological cases may well occur in specific applications. Automatic modification of the batch size limit could be envisaged, whereby the batch limit is dynamically set to match the current application’s needs. At the moment, threads are grouped into batches by locality and indirectly through communication, so that threads created on a common processor are placed onto the same batch. An additional grouping criterion could be based on the frequency of inter-thread communication or rely on object-affinity [13, 14]. Furthermore, the application programmer could be given the opportunity to manually specify viable thread groupings to override the automatic batching arrangements adopted by the scheduler.
The investigations presented here fit into the wider context of a general purpose server-side parallel processing system composed of a cluster of shared memory multiprocessors with high speed user-level CSP communication over Gigabit Ethernet between nodes in the cluster, as well as gigabit speed user-level TCP/IP connectivity to the outside world [15, 16, 5]. Many of the constituent components have already been developed or are at an advanced stage of development.

References

14. S. Abela. Improving fine-grained multithreading performance through object-affinity scheduling. B.Sc. I.T. Final Year Project, Department of Computer Science and Artificial Intelligence, University of Malta, June 2002.
16. S. Busuttil. Integrating fast network communication with a user-level thread scheduler. B.Sc. I.T. Final Year Project, Department of Computer Science and Artificial Intelligence, University of Malta, June 2002.
Author Index

Abela, Charlie, 1
Abela, John, 10
Abela, Patrick, 18
Attard, Malcolm, 21
Bajada, Josef, 27
Busuttil, Steven, 34
Cachia, Ernest, 40
Calleja, Sinclair, 45
Cordina, Joseph, 48
Farrugia, Paulseph-John, 53
Galea, Aaron, 58
Guillauimier, Kristian, 64
Micallef, Mark, 68
Montebello, Matthew, 71
Muscat, Andre', 73
Muscat, Sergio, 80
Pace, Gordon, 84
Rosner, Michael, 92
Spina, Sandro, 99
Spiteri, Kenneth, 103
Staff, Christopher, 106
Talbot, Ingrid, 110
Vella, Kevin, 116