Proceedings of the Sixth Annual

Research Exposition

2009

Proceedings of the Sixth Research Exposition (Research Expo '09)

Fredericton, New Brunswick, Canada

April 8, 2009

UNB Information Technology Centre

Publisher: UNB Information Technology Centre

ISBN 978-1-55131-134-0

Contents

An Embedded Decryption/Decompression Engine using Handel-C

Farnaz Gharibian and Kenneth B. Kent
Reconfigurable Computing Laboratory
Faculty of Computer Science
University of New Brunswick
Fredericton, NB, Canada
{f.gharibian, ken}@unb.ca

Abstract

Speed and security of data streams are two key factors in different areas such as data communication and multimedia. Compression algorithms are applied to data streams to increase their communication speed while encryption algorithms are used for assuring the security of the data transfer. AES and LZ77 are two well known algorithms for data encryption and compression respectively. In this paper we propose a model to implement both algorithms, decryption and decompression, in a Field Programmable Gate Array chip. Such a design must address the issues of optimal resource usage of the FPGA, and balance between the throughput of both algorithms. Handel-C [1] is considered as the specification language for this design.

1 Introduction

The rapid growth of communication data (e.g. audio and video) is powered by faster systems demanding greater speed. To optimize the speed of data transferring in data communications, compression algorithms are developed to reduce the size of the data on the network. On the other hand, security of transferring data has recently become an important issue. Encrypting the data before sending it to the destination is a mechanism for providing data security. Encryption algorithms are developed for this purpose. To benefit both speed and security in working with data streams, encryption and compression algorithms are used together for sending data. By this approach, data is sent with higher speed while keeping the security.

The problem arises when there is a huge amount of encrypted and compressed data that requires decryption and decompression in real time at the destination. The software algorithms for decryption and decomposition lack high performance in real-time processing thus resulting in a delay or jitter appearance in the media. Implementing the algorithms in hardware, however, may significantly improve performance. Moreover, the advantage of parallelizing the algorithms is much more realized if they are implemented in hardware.

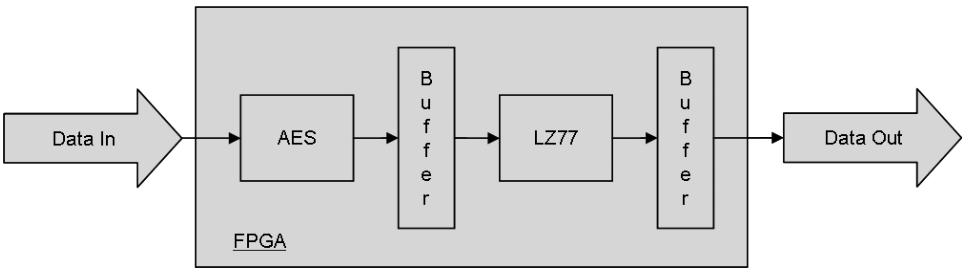


Figure 1. High-level architectural view of decryption/decompression (DecRO) engine

Field Programmable Gate Arrays (FPGAs) have been proven to be very effective and efficient devices on which to implement high performance algorithms [2]. FPGA technology has become powerful, less expensive and more practical for use in real-time applications. FPGAs perform at much faster data rates than their equivalent software implementations because they can do multiple calculations in parallel.

This paper discusses work on a high performance intellectual property (IP) core for real-time data decryption and decompression. By implementing an efficient algorithm for each on a single FPGA, better performance will be achieved while meeting real-time constraints and delivering high quality streamed data (i.e. video and audio).

The resulting decryption/decompression (DecRO) engine [3] design is of interest to a company in the gaming industry. A user selects from an interface the choice of game that they wish to play. Any game selected is loaded from compact flash memory where it resides as encrypted and compressed data. During loading it is given to a software program to prepare it for game play by the user. The bottleneck in this system is the software part that is going to decrypt and decompress the data. This bottleneck is viewed as ”dead time” and is desirable to be minimal to reduce user irritation.

New algorithms in this area are developed continuously. A flexible solution is needed that can be adapted to the changes in these algorithms. The FPGA platform provides a flexible solution to various decryption and decompression algorithms while achieving hardware acceleration needed for computationally intensive processes. This reconfigurable platform also provides the flexible implementation of new decryption and decompression algorithms. Different algorithms can be selected and dynamically downloaded into the platform to reconfigure the hardware based on the needs of the users.

2 Related work

Various hardware implementations of decryption and decompression algorithms have been carried out by other researchers. Huebner et al. [4] published an approach of compressing configuration data using the LZSS compression algorithm at design time and decompressing them with a hardware module implemented on an FPGA during run-time. Li et al. [5] used a lossless compression algorithm in data transmission and storage applications.

A pipelined implementation of AES is proposed in [6, 7, 8]. Rouvroy et al. [9] proposed an AES encryption/decryption design on one FPGA board. Akil et al. [10] developed a hardware implementation of GZIP which is a lossless compression/decompression algorithm. They implemented their architecture on a Xilinx Virtex XVC400 FPGA that uses a PCI bus for data transfer. The results show that the architecture is about two times faster than the software version.

Ou et al. [11] developed an Image Compression Encryption Scheme (ICES). They implemented their

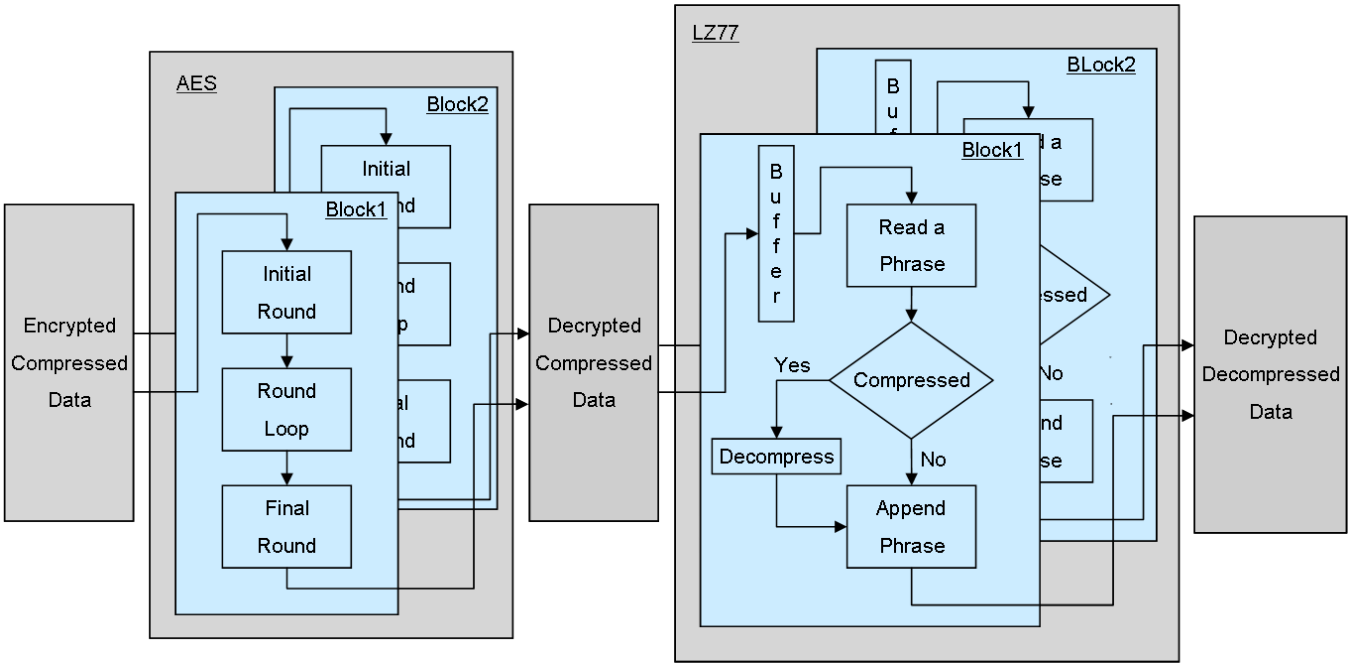


Figure 2. DecRO engine architecture

design in two separate Altera chips and achieved a throughput of 330 Mb/s with a maximum clock rate of 40 MHz.

The proposed model in this paper is different as it performs both the decryption and decompression in a single chip together. By putting both together on a single chip we have a high bandwidth communication link between the components. As well, the choice of block-based algorithms, AES and LZ77, permits pipelining of blocks between decryption and decompression. In this case, decompression can begin while the next block is decrypted. We also intend to make a balance between the implementation of the two algorithms to achieve higher performance. In addition, it will also be developed to support the replacement of the algorithm blocks with newer algorithms for future updates. Encryption and Compression standards are constantly changing with improvements for delivering more security and compaction of data. Designing with the intention to support replacement of these hardware blocks ensures the longevity of the design.

3 Encryption and Compression

Different algorithms are used in encryption and compression systems. In this work, we investigate two specific algorithms; Rijndael algorithm which is known as the Advanced Encryption Standard (AES) for encryption and LZ77 from Lempel Ziv family for compression.

AES was published in 1998 by Vincent Rijmen and Joan Daemen [12] and was originally submitted with the name ”Rijndael”. In 2001, the National Institute of Standards and Technology (NIST) announced the selection of Rijndael as the AES standard. Since then AES has been accepted for encrypting sensitive data streams. AES is a symmetric block cipher that supports different key lengths of 128, 192 or 256 bits. The algorithm makes different transformations on data blocks to encrypt them. For each input block, the algorithm starts with adding the first key to the block. Several iterations of

transformations called rounds will then be performed. Each round is composed of a sequence of four transformations: *ByteSubstitution*, *ShiftRows*, *MixColumns* and *AddRoundKey*. The final step is performing a round without the *MixColumns* transformation.

LZ77 is part of the Lempel-Ziv family of algorithms for lossless data compression. The algorithm was proposed in 1977 by Jacob Ziv and Abraham Lempel [13]. LZ77 is a dictionary-based compression algorithm that uses already processed data as a dictionary. The LZ77 algorithm functions by splitting a sequential input stream into blocks. Each block is parsed by moving a fixed-size window (sliding window) over the data. When a phrase is encountered that has already been in the sliding window, the algorithm attaches a pair of values corresponding to the position of the phrase in the sliding window and the length of the phrase to the output.

4 DecRO

A hardware platform for high-speed processing of decryption and decompression is presented. The high level architectural view of the decryption/decompression engine is shown in Figure 1. The engine is comprised of the AES component, LZ77 component and two buffers, the Intermediate buffer and Output buffer. The AES component is capable of performing decryption of the incoming data stream at a rate that does not provide a significant lag in communication time. This satisfies secure data transfer without negatively impacting the data delivery. The decrypted data goes to the Intermediate buffer to be used by the LZ77 component. LZ77 is a decompression hardware circuit that will allow the information to be transferred at a higher rate since it is compressed. The final decompressed and decrypted data goes to the Output buffer ready to be used by other devices. We make use of pipelining and parallelism in our design to obtain higher performance.

A more detailed design of the engine is shown in Figure 2. AES performs the decryption process in different steps. The AES decryption algorithm operates by applying the inverse of all the transformations described for encryption in reverse order for each data block. In each step, a round is called which contains four particular transformations: *InvByteSubstitution*, *InvShiftRow*, *InvMixColumn* and *addRoundKey* [12]. The Initial *Round* process performs *addRoundKey*. *Round Loop* iteratively calls the mentioned transformations. The number of the iterations is related to the key length. *Final Round* calls *InvByteSubstitution*, *InvShiftRow* and *addRoundKey* transformations. LZ77 decompresses the input data by reading data blocks. Figure 2 shows LZ77 with two block components. In each block, data will be processed sequentially by reading the phrases. If the input phrase has been compressed, the related string is extracted from the sliding window and then appended to the output. The sliding window is a fixed size buffer that keeps the last decompressed data.

Implementation of both decryption and decompression components together while achieving high performance is a challenging issue. We considered pipelining and parallelism in the design of AES and LZ77 components to achieve high performance. However, full pipelining and parallelism requires a large amount of resources. Therefore, there is a trade off between achieving higher performance and using fewer amounts of resources.

Many related works in decryption/decompression hardware design implement only one algorithm in a chip, while in our design synchronization between two different algorithms, AES and LZ77, in one chip is another challenging issue. If AES and LZ77 components do not work with the same speed, buffer overflow/underflow will occur. In a buffer overflow situation, the AES component should pause (or slow down) until LZ77 gets data from the buffer. In a buffer underflow situation, the LZ77 component should

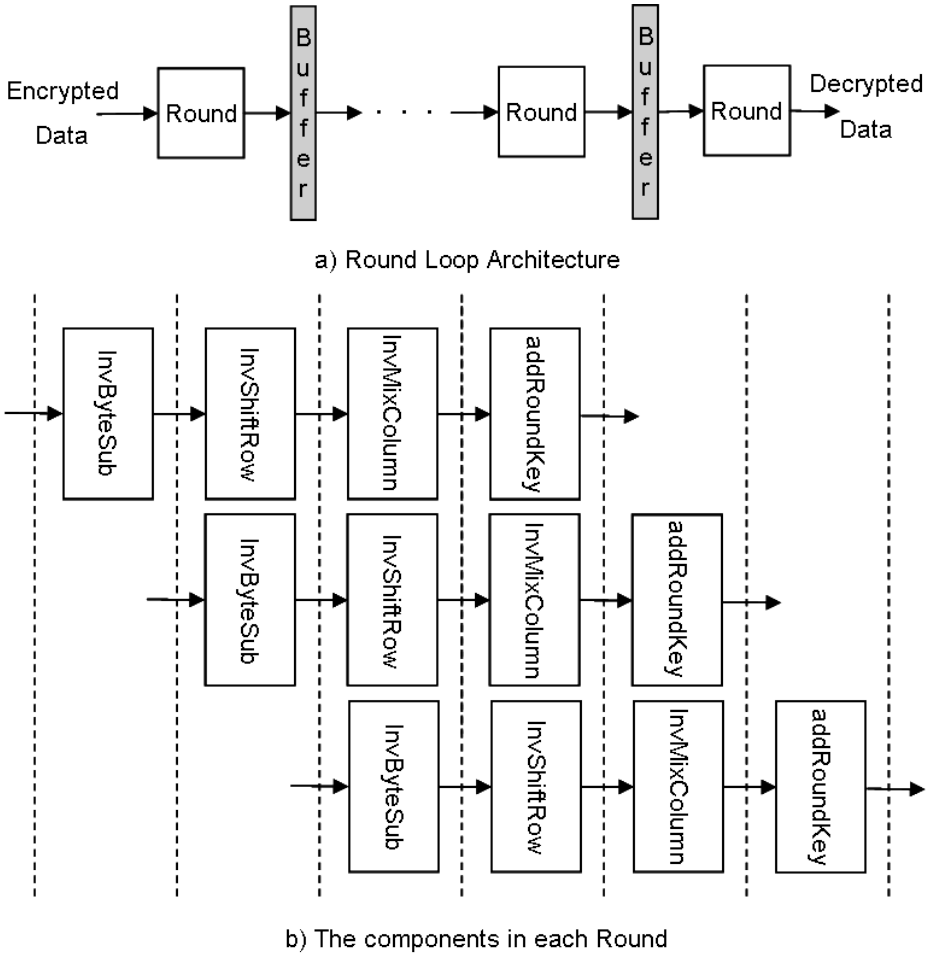


Figure 3. AES design

pause (or slow down) and wait for the AES component to produce data and put it in the buffer.

The AES and LZ77 designs are described in Sections 4.1 and 4.2 respectively. Using parallelism and pipelining in the AES and LZ77 designs increases the performance of each design. There should be a balance in data throughput with the components to ensure no over/underflow occurs with the connecting channel. Avoiding channel contention permits both components to operate at maximum performance without reaching a blocking condition.

4.1 AES Design

The data independency between data blocks is used to parallelize the AES implementation. The design of parallelizing the decryption algorithm in different blocks is shown in Figure 2. Each input data block goes to a different AES block. The number of AES blocks is a factor of improving the performance. Considering two AES blocks shown in Figure 2, the speed of the decryption would be almost twice.

Moreover, a single iteration of the *Round Loop* of each AES block is designed in a component. Up to n components can then be used in a pipeline while n represents the number of iterations in the loop as shown in Figure 3. Therefore, the speed of each block is increased by a factor of n ignoring the

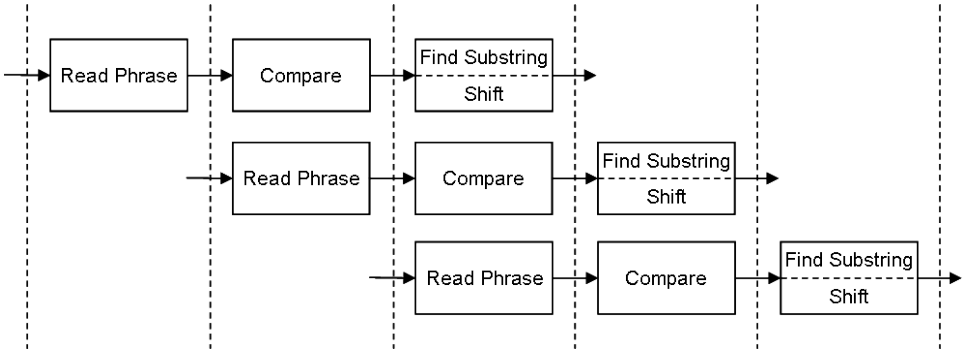


Figure 4. LZ77 design

overheads. The rounds of the *Round Loop* are implemented in separate components (see Figure 3.a). Figure 3.b shows the pipeline design of different transformations that are used in a *Round*. The number of components is related to the key length. For example, 9 components will be used for AES-128 that uses keys with 128 bits length.

4.2 LZ77 Design

Since each data block is individually compressed, the decompression of the different data blocks can be parallelized. The number of block components in LZ77 depends on the size of the block data, the speed of AES and the speed of the device that uses the decompressed data. During the processing of a block, the data stream of another block can be sent to a different component for processing. We are going to use pipelining in the design of each block component in LZ77. Three of the pipeline stages are shown in Figure 4. Stage 1 is reading the phrase from the input buffer of the block. In Stage 2, the phrase is checked to see whether it is compressed. The substring of the phrase is found in Stage 3 and will be added to the Output buffer.

5 Implementation

Handel-C is a high level hardware description language that is based on ANSI-C [1]. Handel-C allows software designers to easily convert their algorithms into a hardware implementation and also allows hardware designers the freedom to easily write functional descriptions of hardware systems. While Handel-C implements only a subset of the ANSI-C standard, it also includes a number of hardware specific constructs to support development of hardware.

We used Handel-C as our specification language for the DecRO engine. Handel-C provides a higher level of specification from the more structural based hardware languages, VHDL and Verilog. Assignments (except signal assignments) and the delay keyword both take one clock cycle to complete. The delay keyword does nothing but consume a clock cycle. It can be useful to break combinational loops or to avoid resource conflicts. The key distinction between Handel-C (for hardware) and ANSI-C (for software) is the use of the *par* keyword to denote multiple operations running in parallel.

We took advantage of parallel programming that is available in Handel-C. Our two modules, Decryption and Decompression work in parallel. We have used a FIFO channel for the communication buffer between these two modules. Figure 5 shows the implemented design of our engine. The two modules,

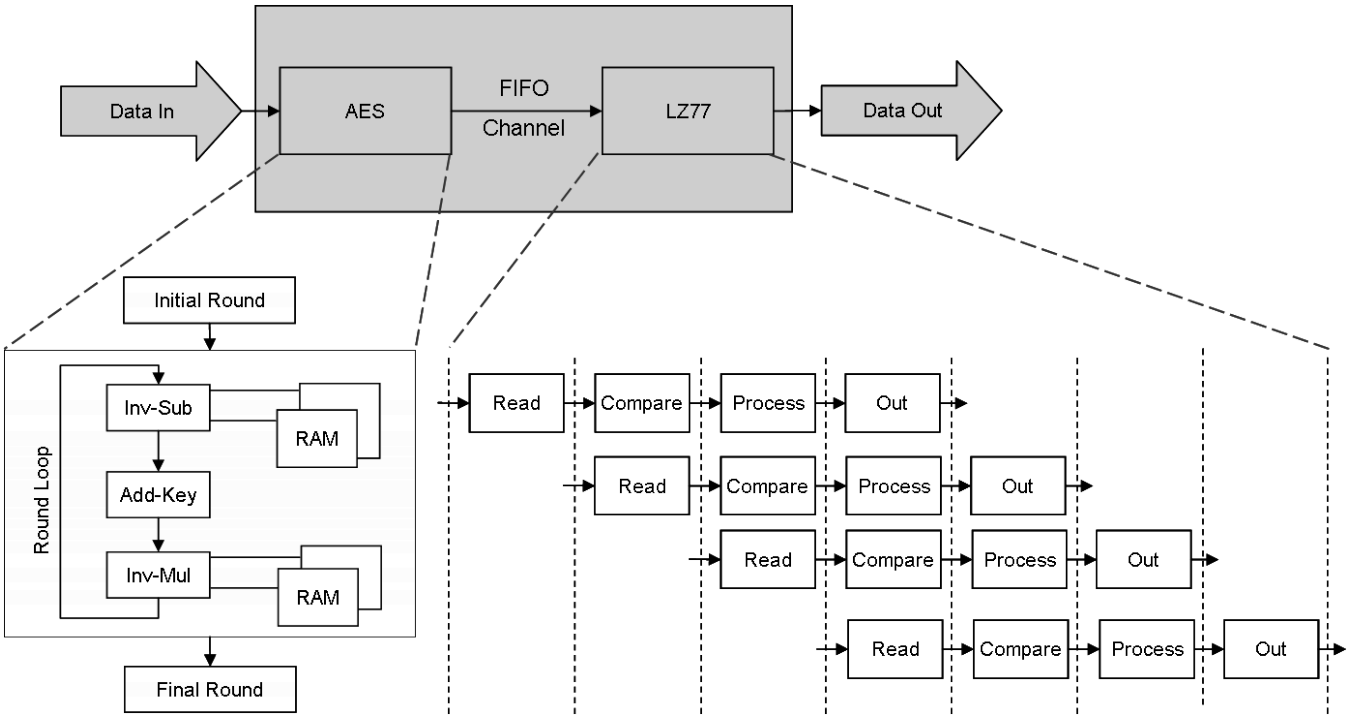


Figure 5. DecRO Implementation

Decryption and Decompression, communicate with each other using a 16-bit FIFO channel. The DecRO engine reads 32 bits of compressed-encrypted data from the input device and sends out 32 bits decrypted-decompressed data to the receiver.

RAMs and ROMs can be implemented directly using the *ram* and *rom* keyword. Specifying the block parameter in conjunction with the *ram* keyword can identify Block RAMs in a target FPGA device. Normal variables are implemented as flip-flops. In the decryption module, we used different rams for storing SBoxes and Galois Field multiplications. To increase our efficiency, dual-port block RAMs are used. By using dual-port block rams, we may access two different locations in a ram at the same clock cycle. Parallel access to the RAM block is considered in the decryption module to improve the speed as shown in Figure 5. There is parallel access to the multiplication and SBox RAM blocks in two modules of the *Round Loop* process.

A pipelined architecture is used in designing the decompression module. Our pipeline has four stages as shown in Figure 5 and is an asynchronous pipeline. Channels are used for communication between stages. Stage one, *"Read"*, gets the data from the interface FIFO channel between the decryption and decompression and puts them in data chunks to send to the next stage. Stage two, *"Compare"*, works on the data and gets the needed information based on if the data is compressed. If the data is not compressed it gets the next character otherwise the offset and length of the data that is compressed. Stage three, *"Process"*, gets the information and adds the character to the sliding window. In the last stage, *"Out"*, decrypted/decompressed data will be sent out in 32 bit chunks.

Table 1. Throughput for different Implementations

Module	**Sequential** (Mbps)	**Parallelism** (Mbps)	**Pipelining/RAM Optimization** (Mbps)
Decryption	22.22	35.90	101.7
Decompression	47.90	80	193.53
DecRO Engine	15.22	26.27	101.36

Table 2. Performance Results

Module	**Throughput (Mbps)**	**Maximum Frequency (MHz)**	**Resource**	
			(# of 4 input LUTs)	**(# of BRAMs)**
Decryption	101.7	102.166	1453	20
Decompression	193.53	205	309	0
DecRO Engine	101.36	108.554	1483	20

6 Simulation Results

Three different implementations are considered for our decryption/decompression engine. Table 1 shows throughput based on megabits per second (Mbps) for the modules in different implementations. Column one shows the results for a sequential design of both decryption and decompression. Column two shows the results after adding parallelism in the design of both the decryption and decompression modules. As can be seen, throughput of the modules is almost increased by a factor of two. The last column which is based on the design shown in Figure 5, the decryption module is improved by adding parallel accessing to RAM blocks and the decompression module is improved by using a pipeline architecture. In the final design, DecRO has reached a throughput that is almost four times of the throughput in the second design.

The implemented DecRO engine is evaluated based on three different parameters: Throughput, Maximum Frequency and Resources. The results shown in Table 2 are based on the final design in Figure 5. Column one shows the throughput of the design. Column two shows the maximum frequency and column three is the consumed resources.

The maximum frequency is calculated by synthesizing VHDL files of the modules using the Xilinx ISE tool targeting the Xilinx Virtex-II Pro chip. VHDL files are created from Handel-C codes using the Celoxica DK tool.

The Throughput(DataRate) is calculated as follows:

$$DataRate = \frac{MaxFrequency * InputBits}{N_{CL} - N_{CF}} \quad (1)$$

Where N_{CF} represents the number of clock cycles for receiving the first output data and N_{CL} represents the number of clock cycles for receiving the last output data.

We measured the consumed resources for each module after place and route using the Xilinx ISE tool targeting the Xilinx Virtex-II Pro XC2VP100 chip. The resource reports are based on the number of 4 input LUTs and Block RAMs used in each module. As it is shown in Table 2, the Decryption module uses more resources than Decompression module.

The DecRO engine throughput and frequency is limited by the decryption module. As the results show, by applying a pipeline design to the decompression module, very high throughput and frequency was achieved. The overall resource usage was one of our key factors. We wanted to implement this module on an FPGA chip where our DecRO engine is not the main function, but instead claims unused resources. The results shows that we have consumed few resources for the design of our engine.

7 Discussion

Our engine has achieved the data throughput of almost 13 MBps. This data rate can be used in conjunction with flash memory without having any significant delay (other than filling the DecRO pipeline) during transferring information while decrypting and decompressing.

Based on the results, two major improvements are as follows:

- The Decryption module should work at a higher frequency since Table 2 shows that the frequency of the decryption module is much lower than the decompression module.
- The Decompression module should work in such a way that is adaptable to the decryption module for achieving the ideal performance for the DecRo engine.

By applying these changes, we can achieve a more powerful DecRO engine that can work with newer faster versions of Flash Memory cards.

8 Conclusion

We implemented DecRO, a high performance decryption/decompression engine as a custom hardware chip. The custom hardware is developed using a Field Programmable Gate Array (FPGA) that allows for rapid prototyping. Handel-C is used as the hardware specification language. AES and LZ77 are considered as the decryption and decompression algorithms that are implemented in the DecRO engine. The performance of the engine will be further increased by using more parallelism and pipelining in the implementation of the algorithms, in progress.

Our goal is to improve the design of DecRO in a way that it can support different decompression and decryption algorithms in a FPGA with high performance. This design can be used in a large area of applications and can support the interchange of different decryption and decompression algorithms.

Acknowledgment

The authors would like to thank Canadian Microelectronics Corporation (CMC) and Natural Sciences and Engineering Research Council of Canada (NSERC) for supporting this research.

References

[1] *Celoxica Ltd.,Handel-C Language Reference Manual*, 2003.

[2] S. D. Brown, R. J. Francis, J. R., and Z. G. Vranesic, *Field-Programmable Gate Arrays*. Kluwer Academic Publishers, 1992.

[3] F. Gharibani and K. B. Kent, "A configurable decryption/ decompression (decro) engine," *Euromicro Conference on Software Engineering and Advanced Applications (SEAA) Work-In-Progress Session*, 2007.

[4] M. Huebner, M. Ullmann, F. Weissel, and J. Becker, "Real-time configuration code decompression for dynamic fpga self-reconfiguration," *18th International Proceedings on Parallel and Distributed Processing Symposium*, 26-30 April 2004.

[5] M.-B. Lin, J.-F. Lee, and G. E. Jan, "A lossless data compression and decompression algorithm and its hardware architecture," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 9, pp. 925–936, September 2006.

[6] N. Nedjah, L. Mourelle, and M. Cardoso, "A compact piplined hardware implementation of the aes-128 cipher," *Proceedings of the Third International Conference on Information Technology: New Generations (ITNG'06)*, 2006.

[7] A. Hodjat and I. Verbauwhede, "A 21.54 gbits/s fully pipelined aes processor," *12th Annual IEEE Symposium on FPGA Field-Programmable Custom Computing Machines(FCCM)*, pp. 308– 309, 20-23 April 2004.

[8] H. Li and J. Li, "A high performance sub-pipelined architecture for aes," *IEEE International Conference on omputer Design: VLSI in Computers and Processors (ICCD)*, pp. 491–496, 2-5 Oct,2005.

[9] G. Rouvroy, F.-X. Standaert, J.-J. Quisquater, and J.-D. Legat, "Compact and efficient encryption/decryption module for fpga implementation of the aes rijndael very well suited for small embedded applications," *Proceedings of the International Conference on Information Technology: Coding and Computing(ITCC'04)*, vol. 02, p. 583, 2004.

[10] M. Akil, L. Perroton, and T. Grandpierre, "Fpga-based architecture for hardware compression/decompression of wide format images," *Journal of Real-Time Image Processing*, vol. 1, pp. 163–170, 2006.

[11] S.-C. Ou, H.-Y. Chung, and W.-T. Sung, "Improving the compression and encryption of images using fpga-based cryptosystems," *Multimedia Tools Applications*, vol. 28, no. 1, pp. 5–22, 2006.

[12] J. Daemen and V. Rijmen, *AES Proposal: Rijndael*. Erewhon, NC: Proton World Intl., March 9, 1999.

[13] J. Ziv and A. Lempel, “A universal algorithm for sequential data compression,” *IEEE Transactions on Information Theory*, vol. 23, no. 3, pp. 337–343, 1977.

Quantifying Process Model Conformance Through Minimal-Cost Approximations

Bruce Hamilton and Liqiang Geng

National Research Council Canada, Fredericton, NB, E3B 9W4
{Bruce.Hamilton,Liqiang.Geng}@nrc.gc.ca

Abstract. In this paper, we provide a solution for identifying the most likely paths in concurrent-process models based on corresponding process instances. The solution is a fixed-parameter tractable heuristic search for workflow (Petri) nets with weighted transitions. The performance of the algorithm was evaluated using eight noisy event logs from [3], where costs were used in a potential conformance measure. The approach allows for several applications towards descriptive analyses of the conformance between a process model and an event log.

Key words: process mining, process conformance, process validation, workflow, approximation.

1. Introduction

Process mining refers to the automated discovery of high-level process models from the use of workflow evidence found in the form of *event logs*. Typically, process mining is used by developers to gain insight into the execution of a process. The approach described in this paper attempts to accurately test the conformance between a process model and a corresponding event log through the most likely sequence of events that the event log could have represented in the model, and calculating the similarity between the sequence and the workflow model.

In the following section, we present a review of related works in the field of conformance checking and process approximation. Section 3 provides the definition of our method for identifying event traces, Section 4 shows an experimental evaluation of the approach, and Section 5 concludes the work done.

2. Related Work

Due to the high-level nature of *process mining*, there is an emphasis placed on the readability and presentation of process models. This emphasis allows for a tendency of neglecting the overall accuracy of the workflow in the model. The balancing act between the accuracy and readability of process models has given rise to a number of useful metrics for measuring the global quality of a process mining algorithm. A useful summary of these metrics is found in [7], where a sizeable collection of process mining algorithms have their discovered models tested against the event logs used. The set of metrics used on the models are grouped into the dimensions of *fitness*, *precision*, *generality*, and *structure*. The approach developed in this paper centers mainly on the *fitness* between a model and a log.

The *fitness* of a model refers to the degree of elements in an event log that can be replayed in the model by following the transitions designated in the model. There have been several metrics proposed to handle the measurement of the fitness of a model, where some are more successful than others. These metrics can be grouped into *token-based*, *trace-based*, and *event-based* categories, where each metric gives an estimate of the rate of correctly parsed instances of its target parameter.

In [6], an analysis of these various fitness measures is performed to offer an accurate summary of the appropriateness of the current methods. In the conclusion of the review, it is stated that the *token-based* approaches in [5,8] have an overly optimistic way of handling fitness, and are unlikely to return a value lower than 0.4 when noise in a log is as high as 90%. The metric used in the *genetic process miner* [3] is said to be too unpredictable at moderate noise levels to be used as a descriptive measure for fitness. The review suggests that for very low levels of noise, the trace based [8] and model level [9] metrics may be appropriate, whereas for broader ranges of noise, the event-based metric [9] would serve best as a conformance measure.

Although the review of conformance measures gives good insight towards choosing a conformance measure, all of the metrics mentioned use boolean values for whether or not a given event or process instance parses correctly (i.e. they do not account for probabilistic bisimulation). A more stochastic approach is described in [2], where Alves de Medeiros et al. describe a method for comparing process models based on their most likely behaviours. This method is useful for cases where two models need to be compared based on a given log, however it does not provide a good measure for testing the conformance of probabilistic models. Instead, the paper refers the reader to [4] for topics in the comparison of probabilistic process models.

In [4], a small set of metrics regarding the bisimulation of Markov processes in discrete and continuous time are described. These metrics describe the "distance" between two probabilistic Markov processes, which could easily be translated to any process model that has directed arcs that are weighted in terms of probability. Approximation is also accounted for in this paper, and other works for determining which process a Markov process shares the most similarity [4].

The metrics and approximations for Markov processes only seem to be tailored to processes where there is a static number of states and transitions between a set of comparable process models. This would cause some difficulty when calculating process models with missing activities, such as in the case where events may be skipped or added during the execution. As well, the methods are only used when comparing two or more models, which could be ineffective for our application, due to the impredictability of process model structure when mining a workflow log.

A more relevant approach was introduced in [1], where comparisons and approximations were made with Markov processes at the event level. The proposed method uses a heuristic search to find the best path in a process model for an event trace that does not necessarily follow the behaviour described in the model. The example process models displayed in the paper were finite state machines, which are not as expressive as Petri nets in terms of concurrency, however the approach can easily be extended to account for the additional models. In the following section, we describe an approach for process instance approximation using a heuristic search where concurrency is present.

3. Our Approach

As stated in the previous section, our approach acts as an extension to the approximation algorithm found in [1] for Markov processes. The key difference is that our algorithm accounts for process models that can represent concurrency (i.e. workflow nets). A workflow net is simply a connected Petri net with one start place and one end place, a much better definition can be found in [3]. The approximation algorithm for our implementation is shown in Figure 1, which makes use of a QueueElement data structure (Figure 2), and insertion and deletion functions (Figures 3 and 4, respectively).

The basic approach used by the algorithm to overcome concurrency is to maintain a list of active places for every branch of the search tree. Rather than attempting to insert only the outgoing states from the current location like in [1], it is necessary to add all possible insertions that would follow a given set of active places in a workflow net. When inserting a new transition to the approximation, we need to enforce that all of the transition's input places are active. Otherwise, the configuration is illegal, and should therefore not be considered as an option.

In terms of an upper bound for running time, the algorithm falls into $O(n^k)$, where n is the length of the process and k is the maximum cost parameter. This is because instances of concurrency in a process model require that any possible sequence of events be legal during execution. In practice, the algorithm seems to work quite well. The running times for approximations of entire logs is shown in the results section (Section 4).

We consider this approach to be useful as a descriptive measure for process-model bisimulation, because it can return precise results for each process instance. For instance, the approach may be used to simply show the average cost for the set of process instances (as shown in the next section), or it could be used to offer insight into common faults in the log or model, much like how cost-based approximations for spelling could offer statistics for common spelling errors. In the following section, we include an example of using the cost as a potential fitness metric, and we show a screenshot of the implementation using a descriptive report for a single process instance.

Fig. 1 (Approximation Algorithm)

```
Algorithm approximate(WorkflowNet N, ProcessInstance P, Integer k): returns QueueElement
    let e be a new QueueElement
    let Q be a new PriorityQueue, ranked in non-decreasing order of c (cost)
    let s_o be the start place of N
    for all transitions t adjacent to s_o:
        enqueue insertion(null, t, P, {s_o}, 0)
        for i from 2 to k:
            if P[i] = t: enqueue(null, t, P(after i), {s_o}, (i-1)*DeleteCost)
    while Q is not empty:
        e = dequeue(Q)
        if e.c > k: return null
        else if e.t_to = null: return e
```

```
    else:
        e.S = e.S - {places before e.t_to} U {places after e.t_to}
        for all s in e.S:
            for all t_s after s:
                enqueue insertion( e.t_to, e.t_s, e.P, e.c )
        enqueue deletion( e.t_to, e.t_to, e.P, e.c )

if e.t_to = null and e.c is at most k:  return e
else:  return null
```

Fig. 2 (Queue Element)

Let a QueueElement E be a tuple $(t_{from}, t_{to}, P, i, S, c)$ *where:*

- t_{from} *is the transition the move is coming from,*
- t_{to} *is the transition the move is going to,*
- *P is the state of the process instance holding a list of events,,*
- *i is the index within the process instance,*
- *S is the set of currently marked places, and*
- *c is the current cost of E*

Fig. 3 (Insertion)

```
Function insertion(QueueElement e): returns QueueElement
    let S_to be the set of places going into t_to
    if S_to is not a subset of S:
        e.c = infinity
    else if e.t_to ≠ e.P[e.i]:
        e.c = e.c + InsertCost
        add e.t_to to e.P at index e.i
    e.i = e.i + 1
    return e
```

Fig. 4 (Deletion)

```
Function deletion(QueueElement e): returns QueueElement
    if e.t_from ≠ null and |e.P| > e.i + 1:
        e.P = e.P - e.P[e.i + 1]
        e.c = e.c + DeleteCost
```

```
else:
    e.c = infinity
return e
```

4. Evaluation

To show how well the algorithm may be suited as a conformance measure, we applied the method to eight logs taken from the sample logs from the ProM website [10]. The eight logs used were taken from the noisy set of logs found in the *GeneticMinerLogs* section, consisting of four logs with 5% noise and four logs with 10% noise. The logs in the section all have designated kinds of noise, based on how tasks are removed, replaced, or added. The only logs used in this study were the logs that include *all* noise types.

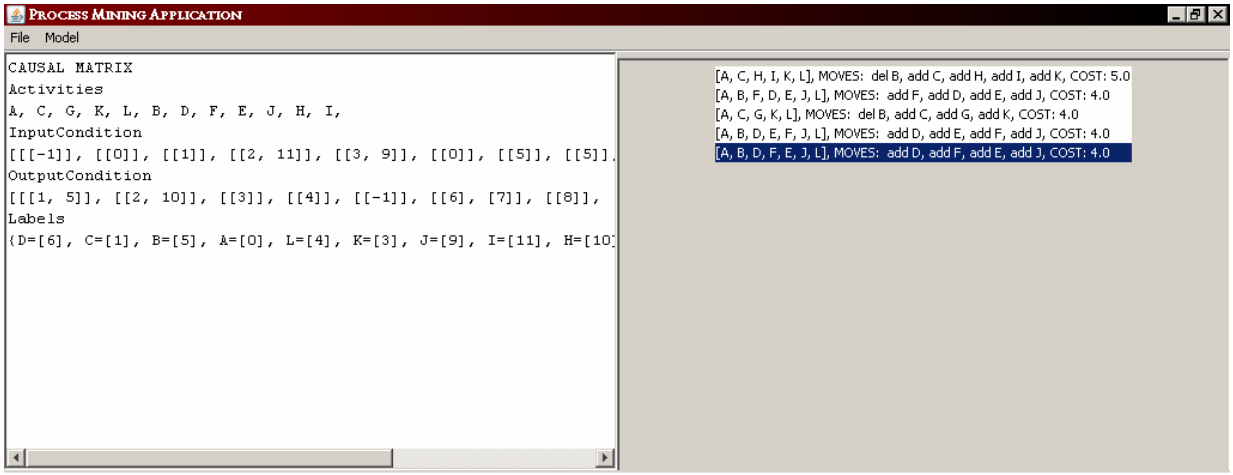
The implementation of the algorithm was programmed entirely in Java. The data structure used for the process model was a *causal matrix*, defined in [3], which was implemented using the class "java.util.Vector" for all the sets found in the causal matrix definition. The causal matrix data structure can easily translate to workflow nets and heuristic nets [3].

The conformance of the logs were tested with the logs' ideal models, as defined in their respective "hn" files, which are all displayed as workflow nets generated by the implementation in Figure 6. Each log was tested with k values from ranging from 1 to 5, and InsertCost and DeleteCosts of 1.0. The entire results are shown in Table 1 on the next page.

In the results table, we record the mean, max, min, and standard deviation of the lowest approximation cost for all process instances omitting the cost values that are greater than k. The number of instances with costs greater than k are shown in the *(>k)* column. The last column shows the time taken for approximations in milliseconds. To give a value of the cost with reference to the size of the log, and to give an example of an appropriate fitness metric, the table includes a new fitness metric given under the column labelled "fitness" in Table 1.

To give an example of how well this metric describes fitness, consider the first log. Not including superficial start and end tasks, the log has a mean of 5.76 events per process instance. With approximately 1.5 corrections for each trace to allow an appropriate fit with the model, we obtain a fitness value of 0.74 as found in the table. To compare this metric with the *proper completion* measure, which simply gives a fraction of traces that properly conform to the model, the value returned by proper completion is 0. Other measures, like the event level metric, could return a value closer to the fitness metric proposed in this paper, but would not be able to return an analysis of what changes would need to be made to the log or model. An simple example of such a response is shown in Figure 5, where a screenshot of the implementation is shown displaying an approximation for a given process instance, including the appropriate actions for modifying the trace.

Fig. 5: A visual display of the approximation of process instance, "A,B,L".


PROCESS MINING APPLICATION
File
Model
CAUSAL MATRIX
Activities
A, C, G, K, L, B, D, F, E, J, H, I,
InputCondition
[[[-1]], [[0]], [[1]], [[2, 11]], [[3, 9]], [[0]], [[5]], [[5]],
OutputCondition
[[[1, 5]], [[2, 10]], [[3]], [[4]], [[-1]], [[6], [7]], [[8]],
Labels
{D=[6], C=[1], B=[5], A=[0], L=[4], K=[3], J=[9], I=[11], H=[10]
[A, C, H, I, K, L], MOVES: del B, add C, add H, add I, add K, COST: 5.0
[A, B, F, D, E, J, L], MOVES: add F, add D, add E, add J, COST: 4.0
[A, C, G, K, L], MOVES: del B, add C, add G, add K, COST: 4.0
[A, B, D, E, F, J, L], MOVES: add D, add E, add F, add J, COST: 4.0
[A, B, D, F, E, J, L], MOVES: add D, add F, add E, add J, COST: 4.0

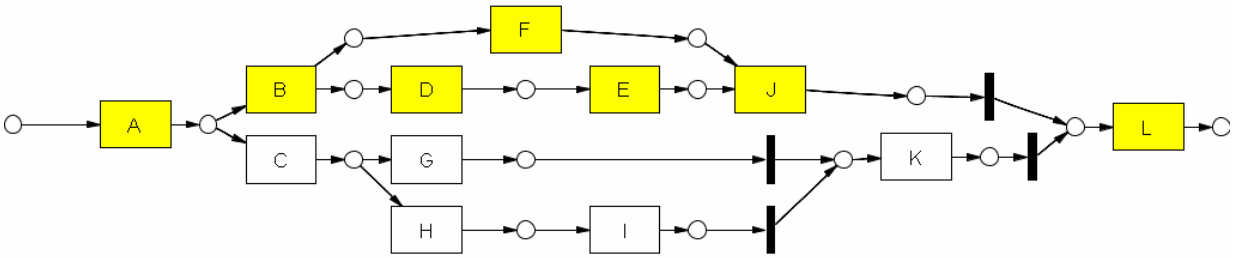
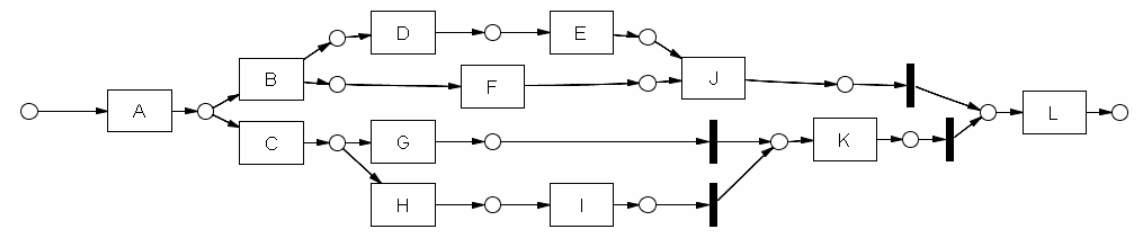

A
B
C
D
E
F
G
H
I
J
K
L

Table 1: Experimental Results

Model	Noise	k	Mean	Std dev	Max	Min	(>k)	Fitness	Time (ms)
A12	5.00%	5	1.53	1.42	4	0	0	0.735	40.60
		4	1.53	1.42	4	0	0	0.735	11.99
		3	0.82	1.60	2	0	3	0.765	9.02
		2	0.82	1.24	2	0	3	0.796	5.61
		1	0.24	1.31	1	0	8	0.878	4.01
	10.00%	5	1.71	1.30	4	0	0	0.698	17.79
		4	1.71	1.30	4	0	0	0.698	15.51
		3	1.00	1.49	2	0	5	0.730	13.35
		2	1.00	1.14	2	0	5	0.761	8.59
		1	0.29	1.34	1	0	15	0.855	5.79
BN1	5.00%	5	2.00	1.91	5	0	0	0.942	1410.71
		4	1.62	1.96	4	0	1	0.944	865.94
		3	0.38	2.41	2	0	5	0.955	219.62
		2	0.38	1.79	2	0	5	0.966	78.11
		1	0.23	1.32	1	0	6	0.980	16.62
	10.00%	5	1.90	2.22	5	0	4	0.921	4974.37
		4	1.55	2.12	4	0	6	0.928	1272.91
		3	0.72	2.32	3	0	12	0.940	275.18
		2	0.41	1.97	2	0	15	0.956	100.56
		1	0.28	1.40	1	0	17	0.974	20.32
Herbst 3.4	5.00%	5	0.35	0.83	4	0	0	0.975	11.87
		4	0.35	0.83	4	0	0	0.975	11.30
		3	0.25	0.84	2	0	1	0.976	12.11
		2	0.25	0.73	2	0	1	0.978	10.62
		1	0.10	0.68	1	0	4	0.985	9.62
	10.00%	5	0.58	0.93	4	0	0	0.955	14.66
		4	0.58	0.93	4	0	0	0.955	14.44
		3	0.50	0.93	3	0	1	0.956	15.60
		2	0.44	0.87	2	0	2	0.959	14.75
		1	0.20	0.83	1	0	8	0.972	11.96
Herbst 6.37	5.00%	5	0.10	0.54	4	0	0	0.993	61.91
		4	0.10	0.54	4	0	0	0.993	78.69
		3	0.04	0.55	2	0	2	0.994	69.57
		2	0.04	0.44	2	0	2	0.995	34.95
		1	0.01	0.36	1	0	4	0.997	33.70
	10.00%	5	0.24	0.76	4	0	0	0.983	86.24
		4	0.24	0.76	4	0	0	0.983	91.25
		3	0.16	0.76	3	0	3	0.984	79.97
		2	0.14	0.67	2	0	4	0.986	40.18
		1	0.04	0.58	1	0	11	0.992	35.95

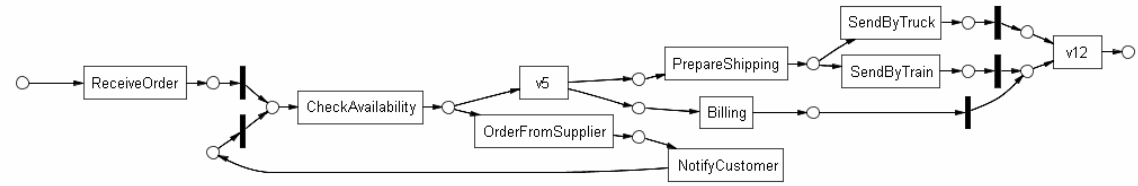
Fig. 6: The test models used in the experiment.

(a) A12

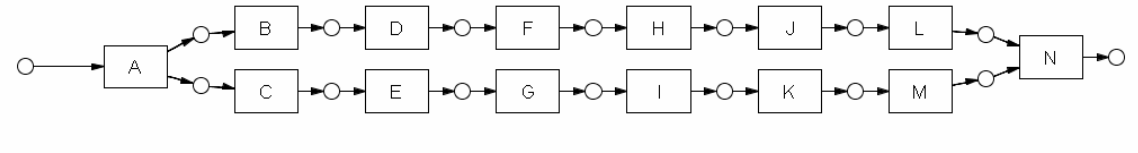


(b) BN1

(c) Herbst Fig 3.4



(d) Herbst Fig 6.37



5. Conclusion

In summary, we have developed a working algorithm for discovering the most probable sequence of transitions to be taken in a process model, given a process instance. The algorithm has been shown to be applicable as a basic metric for determining the conformance between a process model and an event log, and has been proposed as a highly descriptive way for determining common faults in a log or model.

Further applications for this approach could involve a trace-by-trace monitoring of a process execution, where exceptions are thrown when the execution deviates by *k* operations. This could allow for monitoring workflow in an enterprise with a reasonable amount of allowance for different actions, while providing appropriate feedback to the administrator when exceptions are raised.

The algorithm used in this implementation provides a good beginning for the use of approximations in concurrent process models, but it has yet to be optimized for performance. Future work on this approach should involve the refinement and specialization of the implementation.

[1] Jonathan E. Cook and Alexander L. Wolf. Software process validation: quantitatively measuring the correspondence of a process to a model. *ACM Transactions on Software Engineering and Methodology*, 8:147–176, 1999.

[2] A. K. Alves de Medeiros, W. M. P. van der Aalst, and A. J. M. M. Weijters. Quantifying process equivalence based on observed behavior. *Data Knowl. Eng.*, 64(1):55–74, 2008.

[3] A.K. Alves de Medeiros. *Genetic Process Mining*. PhD thesis, Eindhoven University of Technology, 2006.

[4] Josée Desharnais, Vineet Gupta, Radha Jagadeesan, and Prakash Panangaden. Metrics for labelled markov processes. *Theor. Comput.Sci.*, 318(3):323–354, 2004.

[5] A. Rozinat and W. M. P. van der Aalst. Conformance checking of processes based on monitoring real behavior. *Inf. Syst.*, 33(1):64–95, 2008.

[6] A. Rozinat, M. Veloso, and W. M. P. van der Aalst. Evaluating the quality of discovered process models. In *W. Bridewell, T. Calders, A.K. de Medeiros, S. Kramer, M. Pechenizkiy, L. Todorovski (Eds.), Proceedings of Induction of Process Models*, pages 45–52, Antwerp, Belgium, 2008.

[7] A. Rozinat, A.K. Alves de Medeiros, C.W. Günther, A.J.M.M Weijters, and W.M.P. van der Aalst. The need for a process mining evaluation framework in research and practice. In *Business Process Management Workshops*, pages 84–89, Eindhoven, Netherlands, 2008.

[8] A. Rozinat, M. Veloso, and W.M.P. van der Aalst. Using Hidden Markov Models to Evaluate the Quality of Discovered Process Models. Extended Version. BPM Center Report BPM-08-10, BPMcenter.org, 2008.

[9] A. J. M. M.Weijters and W. M. P. van der Aalst. Rediscovering workflow models from event-based data using little thumb. *Integr. Comput.-Aided Eng.*, 10(2):151–162, 2003.

[10] ProM Official Website. URL = http://prom.win.tue.nl/tools/prom/

A Methodology for Rapid Optimization of HandelC Specifications

Joey C. Libby
Faculty of Computer Science
University of New Brunswick
Fredericton, New Brunswick, Canada
g6x2d@unb.ca

Kenneth B. Kent
Faculty of Computer Science
University of New Brunswick
Fredericton, New Brunswick, Canada
ken@unb.ca

Abstract

Utilizing high level hardware description languages for the creation of customized circuits facilitates the rapid development and deployment of new hardware. While hardware design languages increase the speed at which hardware can be developed, creating hardware designs that are both efficient in resource usage and processing speed can be time consuming and require much experience. This problem is compounded more by the long design cycle times that are introduced by the long compilation and synthesis times that are required to translate a high level hardware description language to a circuit. This problem is addressed by performing some of the optimizations automatically, pre-synthesis, reducing the total number of synthesis cycles that are required, saving much development time.

1. Introduction

High level hardware description languages provide a means for new hardware designers to enter the field of hardware design as well as providing a tool for experienced hardware designers to rapidly develop and prototype new technologies. One problem that can plague novice and experienced designers alike is the creation of hardware designs that both perform the task they are designed for quickly as well as utilize available resources as efficiently as possible.

Optimizing hardware designs can be a tedious and time consuming process as possible optimizations are made to a design and then are tested. This cycle of design and testing includes the compilation and synthesis of the hardware design which can be extremely time consuming. Large designs can take hours or even days to compile and synthesize, leading to large amounts of wasted downtime in the development process.

One possible solution to this problem is to perform some of these optimizations automatically, allowing the designer to eliminate a portion of the compilation and synthesis cycle, thus reducing the amount of downtime and speeding the development cycle of new hardware.

This work discusses the changes that will be made to the current hardware design cycle and how these changes will affect the speed at which development of hardware is completed. The methodology will then be tested by applying a tool which automatically identifies simple parallelism in Handel-C hardware designs to several hardware design projects.

2. Background

This section will discuss the background information on HandelC and the automatic extraction of parallelism that is applicable to this paper.

2.1 HandelC

HandelC [1] is a high level hardware description language that bears much resemblance to the ANSI C programming language. While HandelC is very similar to ANSI C in many respects, there are some major differences between the two languages. HandelC does not support the entire ANSI C specification, removing support for some software constructs, most importantly support for runtime recursion is absent from HandelC. HandelC, along with support for a subset of the ANSI C specification, includes extra support useful for hardware description. Included in this extended support are constructs for input and output, communications, and control flow constructs for controlling the concurrency of a design. Concurrency in a HandelC program is defined by using the `par {}` and `seq {}` statement blocks. Sequential instructions wrapped in a `par {}` statement will be executed concurrently, while statements wrapped in a `seq {}` statement will be forced to execute sequentially. Example 1 shows `par` and `seq` statements in a simple HandelC design.

```
int 8 a,b,c,d,e,f,g,h;
a = 1; b = 2; c = 3; d = 4;
par {
        d = a + b;
        e = c + d;
}
seq {
       f = d+e;
       g = d*e;
}
```

Example 1: Example of `par` and `seq` Statements

The existence of the parallel and sequential block keywords provides the facilities necessary for a designer or an automated tool to easily exploit parallelism without the need of generating control logic to do so.

2.2 Automated Extraction of parallelism

Identification of simple parallelism, that is sequential blocks of hardware code that can be executed in parallel, can have a huge impact on the performance of the hardware system that is being designed. The tool that will be used to apply optimizations for the purpose of this paper can be found in [2].

Given a HandelC source file, this tool is capable of parsing and extracting simple parallelism from the source file. This information is then relayed to the hardware designer who can implement the proposed changes in order to build a more optimized version of the original hardware design.

Figure 1 shows an overview of the operation of the automated parallelism extraction tool. The tool operates by taking, as input, a HandelC hardware definition file. From this source file the tool creates an abstract syntax tree, annotated with additional information that is required to compute the dependency graph from the source file. Upon completion of the syntax tree, it is used to generate a dependency graph structure for the hardware design. This dependency graph structure is then used to determine which individual lines of source can potentially be executed in parallel. Currently the tool then applies a greedy algorithm which builds maximum size parallel blocks from the remaining available lines of source code. This approach generates large parallel blocks which in turn reduce the overall run time of computations on the hardware.

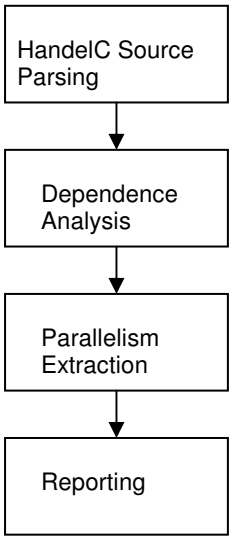


Figure 1: Automated Parallelism Extraction

Once the tool has determined where parallel blocks may be added to the source hardware design, it produces a report for the designer which details the necessary modifications that must be performed in order to exploit the available parallelism. Table 1 illustrates the report output of the tool.

Source Line#	Source	Action
*	Par {	Add
1	Statement 1	None
2	Statement 2	None
3	Statement 3	None
7	Statement 7	Move -
8	Statement 8	Move –
*	}	Add
4	While	Move +
5	Statement 5	Move +
6	Statement 6	Move +
9	Statement 9	Move +

Table 1: Tool Report

In testing, this tool has shown that it is capable of finding, on average, 78% of the simple straight line parallelism that exists in a hardware design. This makes this tool well suited for the purpose of demonstrating the effects of modifying the hardware design cycle as proposed in this paper.

3. Proposed Hardware Design Cycle

The design process shown in Figure 2 outlines the common method for performing hardware design using a high level hardware description language. The hardware design process begins much like that of a software design, by determining the requirements of the project. Following the requirements analysis, the hardware specification can be written. Some important requirements for a hardware system include logic usage (design size), power consumption, design speed and maximum throughput. Upon completion of the hardware specification, the specification must be synthesized into a logic network. This step of logic synthesis is time consuming and can contribute a large amount of downtime to a complex project. Following the synthesis of a hardware specification, the specification is tested in simulation to determine if the implemented functionality is correct. If problems are found during the testing phase they must be corrected, and the corrected hardware specification must be re-synthesized. If the implementation is correct, it can now be benchmarked to determine if the performance of the hardware specification meets the requirements. If the specification does not meet the requirements laid out in the requirements analysis phase the design must be improved through various methods such as adding concurrency or pipelining. Following the design improvements stage, the new specification must be re-synthesized, tested and benchmarked. These phases are repeated until a specification that meets all of the requirements is created. This synthesized specification can then be used to generate a physical device such as an ASIC or FPGA.

The problem with this design flow, is its heavy reliance on the time consuming synthesis process. In this design flow synthesis is required to determine if the design meets many of the design requirements, such as power consumption, speed, logic usage and throughput. The need to iterate through a synthesis process repeatedly increases the amount of time that must be spent in the design and implementation of a hardware system. The proposed methodology, shown in Figure 3, will allow

the creation of an optimized hardware specification while minimizing the amount of time that must be spent in the synthesis process.

The proposed methodology improves on the current flow for hardware design by adding the optimization sub-flow and restructuring the remainder of the design flow. The need to identify concurrency manually is removed. Designs are made concurrent and pipelined automatically by the hardware synthesis tools. Utilizing this automated optimization step allows much of the design flow to be moved to a pre-synthesis stage, minimizing the number of times synthesis must be performed on a given hardware design.

The remainder of this section will discuss the different stages in the new optimization design flow methodology and discuss issues that must be dealt with for each stage.

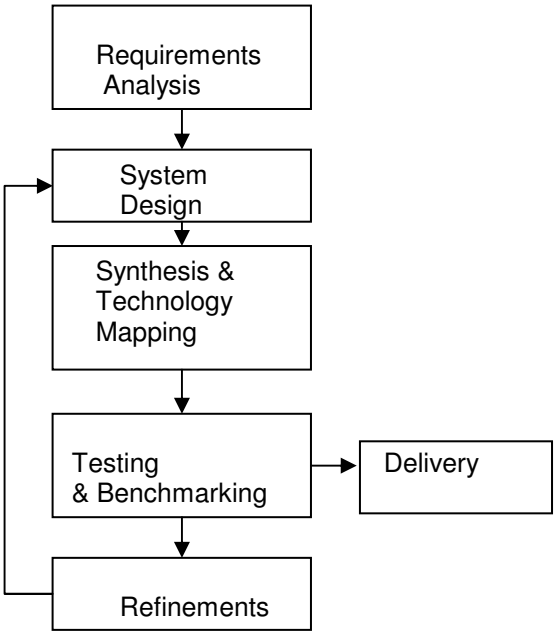


Figure 2: Hardware Design Process

3.1 Optimization Sub-Flow

The optimization sub-flow, Figure 3, allows most of the optimization work, normally done manually post-synthesis, to be completed pre-synthesis using automated compiler optimization. Each of the individual stages in the optimization sub-flow will be described in detail.

3.1.1 Resource Estimation

Much research has been completed in the area of resource estimation for hardware specifications [3,4,5]. Resource estimation allows a hardware designer the flexibility to quickly estimate how many resources are being consumed by a given hardware design. This capability is vitally important to the improved hardware design flow because it allows much of the design flow to be moved to the pre-synthesis partition of the hardware design flow. The resource estimation step in the optimization flow must be performed first in order to establish a baseline set of resource estimations on which the

optimizations will be judged. This will allow further stages in the optimization flow to evaluate the impact of given optimizations on the hardware design. The baseline resource usage, combined with the systems requirements will also provide a target for the optimizations. The resource estimation is performed at two different points in the design flow, the first being the baseline resource estimation, and the second after performing optimizations. The second pass of resource estimation is necessary to determine the impact of changes made during the optimization steps. This post-optimization estimation can then be compared to the pre-optimization estimation as well as the design requirements in order to determine how to proceed.

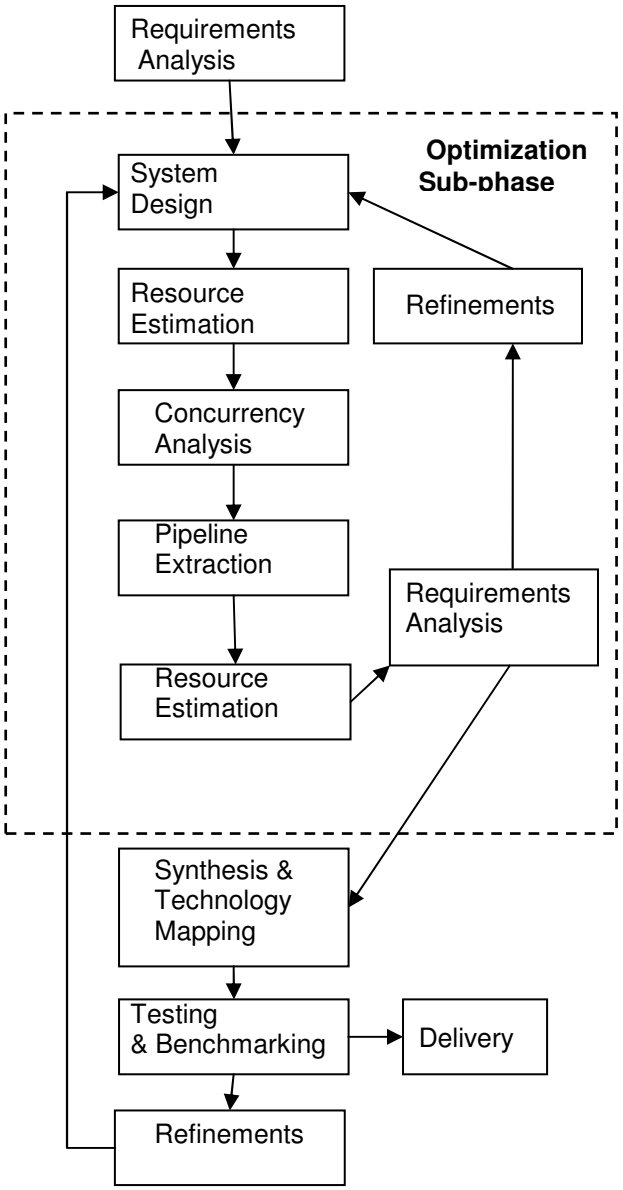


Figure 3: Proposed Hardware Design Cycle

3.1.2 Identification of Parallelism

In this stage of the design flow, concurrency is extracted from the hardware specification. Much research on identification of concurrency has been performed in the software field, most of which is directly applicable to high level hardware description languages. Section 2.2 discusses the automated parallelism extraction tool that will be used for the purpose of testing for this paper.

3.1.3 Pipeline Inference

In this stage of the design flow, a pipelined data path is extracted from the high level hardware description. The pipeline(s) will be extracted using dependency data collected during the concurrency extraction phase. This data will be used to determine if sufficient parallel blocks exist within the design to form a pipelined data path. There have been several pieces of research work completed that attempt to address several of the problem domains within automated pipeline inference. Some of the problems addressed in this research include the problem of scheduling [6], pipeline optimization [7], high level pipeline synthesis [8], pipelining of DSP applications [9] and finally vectorization of pipelined datapaths [10]. The approach to pipeline inference to be used in this work will differ from the previous works in that in [11] the pipelining process is not constrained to the high level hardware description. The proposed approach will allow all modification to the hardware description to be completed at the high level, changing the structure of the description itself, instead of generating lower level pipelined structures such as netlists. The remainder of this section will discuss some of the major issues that must be addressed when attempting to automatically extract a pipelined datapath from a high level hardware description.

4. Testing the Modified Design Cycle

In order to demonstrate the merit of the modified hardware design cycle it will be necessary to test the effects of this design cycle on real world hardware design projects. This section will discuss how this testing will be performed as well as the two test case projects which were optimized using both the standard hardware design cycle as well as the proposed hardware design cycle.

4.1 Testing Methodology

Currently tools do not exist for several of the stages in the modified hardware design process. Figure 4 shows the availability of tools required for this methodoly. While a tool exists that allows for the automated extraction of parallelism within a HandelC hardware definition, tools do not currently exist that allow for the extraction of a pipelined data path or for performing pre-synthesis resource estimation on a design. Without these tools it will still be possible to explore the impact of moving more of the design process to the pre-synthesis stage, removing the need for multiple iterations through the modification and compilation/synthesis stages of the design cycle.

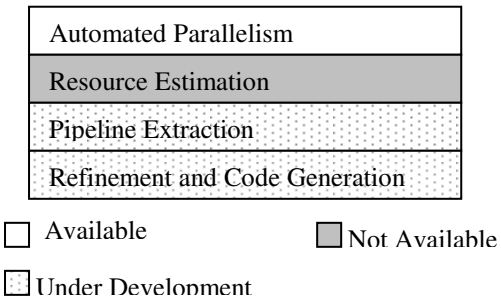


Figure 4: Tool Availability

For the purpose of the two test cases that will be discussed in the coming sections, the projects were completed first using no special tools and using the standard hardware design cycle. A non-optimized version of the project was created first, followed by a hand optimized version of the design. These hand optimized designs, along with the time that was needed to complete them will be used as benchmarks alongside the versions that were optimized by the automated tool using the new hardware design cycle.

4.2 Case Study 1: AES/LZ77 Engine

The first test case for the modified hardware design cycle that will be examined is a project that utilized a combination of the LZ77 compression algorithm in conjunction with the AES encryption algorithm to produce a hardware device which is capable of rapidly performing a sequence of decompression and decryption on a stream of data.

LZ77 is part of Lempel-Ziv family algorithms for lossless data compression by Jacob Ziv and Abraham Lempel [12]. LZ77 is a dictionary-based compression algorithm that uses already processed data as a dictionary. The LZ77 algorithm functions by splitting a sequential input stream into blocks. Each block is parsed by moving a fixed-size window (sliding window) over the data. When a phrase is encountered that has already been in the sliding window, the algorithm attaches a pair of values corresponding to the position of the phrase in the sliding window and the length of the phrase to the output.

The AES algorithm was published in 1998 by Vincent Rijmen and Joan Daemen [13] and was originally submitted with the name ”Rijndael”. In 2001, the National Institute of Standards and Technology (NIST) announced the selection of Rijndael as the AES standard. Since then AES has been accepted for encrypting sensitive data streams. AES is a symmetric block cipher that supports different key lengths of 128, 192 or 256 bits. The algorithm makes different transformations on data blocks to encrypt them. For each input block, the algorithm starts with adding the first key to the block. Several iterations of transformations called rounds will then be performed. Each round is composed of a sequence of four transformations: ByteSubstitution, ShiftRows, Mix-Columns and AddRoundKey. The final step is performing a round without MixColumns transformation.

Upon completion of the hardware design which featured a tightly coupled LZ77 decompression engine and AES decryption engine, the design was manually optimized. The design was carefully analyzed in order to identify any potential areas where parallelism could be exploited, and trough a process of iterative refinement, these parallel blocks were added and tested to ensure a positive impact

on the overall system. When this phase of design was completed it was found that the system overall contained 18 parallel blocks. While this was a positive outcome, and resulted in a much more efficient hardware design, the amount of time which was required, approximately 24 hours, may have been prohibitive under most circumstances, especially when rapid prototype development is required.

Following the completion of the manual optimization of the LZ77/AES engine, the modified hardware design cycle was applied to an un-optimized version of the design. The automated parallelism extraction tool was ran once on the source file, providing a report of all of the identified parallel blocks that may be exploited. In this case, due to the relatively small size of the source file, the tool provided nearly instantaneous feedback, identifying 78% of the parallel blocks that were identified in the manually optimized version. These optimizations were then performed to the source file, which was then compiled, synthesized and tested to ensure that the modifications were correct.

In this case the modified hardware design cycle proved to reduce the amount of time required to apply optimizations to a design by a large factor. Dozens of iterations through the compilation and synthesis steps were eliminated, and the entire process required only one iteration through the design cycle. While the automated tool was unable to identify all of the parallelism that existed in the source design, it was able to identify the majority of available parallelism. This, when coupled with the large amount of time saved in the design process, supports the idea that moving more of the design cycle to the pre-compilation and synthesis stage can have a large positive impact on the development times of hardware.

4.3 Case Study 2: Irreducible Polynomials over GF(3)

This test case is created based on a previous project in which a co-designed hardware solution for the software algorithm in [14]. This solution explored migrating only part of the software computation into hardware, and while successful, it suffered from low performance due to communications between the hardware and software partitions [15].

In order to alleviate the performance degradation caused by communications between the hardware and software in the co-designed system, as well as the low performance of the general purpose processor, a full implementation was created in hardware [16]. This implementation was written in HandelC which allowed the hardware implementation to very closely mimic the software algorithm wherever possible.

This project was first completed without attempting to add any optimizations to the project. The C source code was directly ported to HandelC, with the appropriate modifications for the hardware environment. Following this, the design was optimized manually by analyzing the hardware design and determining where parallel blocks existed. These blocks were then added and tested. Due to the relatively small size of this project, the optimization phase was relatively short, encompassing approximately 8 hours of optimizing and testing. In total, 17 parallel block optimizations were found during the manual optimization of the project.

Following completion of the manually optimized version of the GF(3) hardware design. The un-optimized hardware design was optimized using the modified hardware design cycle. The un-modified source was analyzed by the automated parallelism tool, which in this case found all of the manually identified parallel blocks. Again the tool provided nearly instantaneous feedback, cutting development time from what was 8 hours when manually identifying parallel blocks to a few seconds for the automated tool.

The result from this case study supports the merit of the modified hardware design cycle for shortening development time of hardware designs.

5. Conclusion

While the full impact of using the modified hardware design cycle will not be known until the remaining tools are completed, the case studies that were explored in this paper show that the modified hardware design cycle as proposed in this paper will have a positive impact on the design times required to complete an optimized hardware design.

6. Future Work

Much work must still be completed in order to explore fully the proposed hardware design cycle. This work will be centered on the development of a tool that is capable of analyzing a HandelC source file and extracting from it an optimized pipelined data path. It is hoped that combining both automatic parallelism extraction with automated pipeline extraction that a sufficient level of optimization can be achieved without any direct input from the hardware designer.

In addition to automated parallelism extraction, another key component to the proposed design cycle is the resource estimation tool. A resource estimation tool will allow the automated parallelism and pipeline extraction tools to perform iterative optimizations, targeting a specific goal for resource usage, without the need to run time consuming compilation and synthesis.

Finally, a tool for performing the analysis of requirements as compared to the resource estimates and a code generator for applying proposed changes automatically will also be needed to remove the entire optimization phase from the control of the designer. This will allow the optimization phase to run iteratively multiple times in order to determine the best possible configuration of parallel blocks and the pipelined data path.

References

[1] Agility Design Solutions, HandelC Reference Manual, Website: www.agilityds.com, accessed September 2008.

[2] J. C. Libby, F. Gharibian, and K. B. Kent, Automatic Identification of Parallelism in Handel-C, 11th Euromicro Conference on Digital Systems Design, pp. 660-664, September 2008.

[3] C. Brandolese, W. Fornaciari, and F. Salice, An Area Estimation Methodology for FPGA Based Designs at System-level, Proceedings of Design Automation Conference, 2004.

[4] Rolf Enzler, Tobias Jeger, Didier Cottet, and Gerhard Troster, High-level area and performance estimation of hardware building blocks on fpgas, In FPL ’00: Proceedings of the The Roadmap to Reconfigurable Computing, 10th International Workshop on Field Programmable Logic and Applications, 2000.

[5] D. Kulkarni,W.A. Najjar, R. Rinker, and F.J. Kurdah, Fast Area Estimation to Support Compiler Optimizations in FPGA-based Reconfigurable Systems, Proceedings IEEE Symposium on Field Programmable Custom Computing Machines, 2002.

[6] P. Arato, Zoltan A dam Mann, and Andras Orban, Time-Constrained Scheduling of Large Pipelined Datapaths, Journal of Systems Architecture, 2005.

[7] D.J. Mallon, and P.B. Denyer, A New Approach to Pipeline Optimisation, Proceedings of the European Design Automation Conference, 1990.
[8] Y. Hsu and Yuang-Long Jeang, Pipeline Scheduling Techniques in High-Level Synthesis, ASIC Conference and Exhibit, 1993.
[9] Hong-Shin Jun and Sun-Young Hwang, Design of a Pipelined Datapath Synthesis System for Digital Signal Processing, IEEE Transations on Very Large Scale Integration, 1994.
[10] W. Weinhardt and W. Luk, Pipeline Vectorization, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2001.
[11] N. Park and A.C. Parker, Sehwa: a Software Package for Synthesis of Pipelines from Behavioral Specifications, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1998.
[12] J. Ziv and A. Lempel. A Universal Algorithm for Sequential Data Compression. IEEE Transactions on Information Theory, 23(3):337–343, 1977.
[13] J. Daemen and Vincent Rijmen. AES Proposal: Rijndael. Proton World Intl., Erewhon, NC, March 9, 1999.
[14] G. Lee and F. Ruskey, Listing All Irreducible and Primitive Polynomials in GF(3),. Technical Report (University of Victoria, Canada), 2006. Unpublished.
[15] K. Kent, B. Iaderoza and M. Serra. Codesign of a Computationally Intensive Problem in GF(3), International Workshop on Rapid System Prototyping pp. 10-16, May 2007.
[16] J. C. Libby, K. B. Kent, and J. P. Lutes, A Handel-C Implementation of a Computationally Intensive Problem in GF(3), International Conference on Advances in Electronics and Micro-electronics, pp. 36-41, October 2008.

A Handel-C Implementation of a Computationally Intensive Problem in GF(3)

Joey C. Libby, Jonathan P. Lutes, and Kenneth B. Kent
Faculty of Computer Science
University of New Brunswick
Fredericton, New Brunswick, Canada
g6x2d@unb.ca, f9dz2@unb.ca, ken@unb.ca

ABSTRACT

Computing the irreducible and primitive polynomials under GF(3) is a computationally intensive task. A hardware implementation of this algorithm should prove to increase performance, reducing the time needed to perform the computation. Previous work explored the viability of a co-designed approach to this problem and this work continues addressing the problem by moving the entire algorithm into hardware. Handel-C was chosen as the hardware description language for this work due to its similarities with ANSI C used in the software implementation.

1. INTRODUCTION

The performance of many software systems can be improved by the creation of custom hardware circuits that are capable of performing some or all of a software systems processing in a native hardware environment [8,9,10,11]. One major reason that software is implemented in hardware is the core features that a hardware implementation offers a system designer. The most important of these features is the inherent parallelism that is found in hardware systems such as Field Programmable Gate Arrays (FPGA).

The work presented in this paper is a continuation of work started in [1] and centers around the completion of migrating a software system for the computation of irreducible and primitive polynomials over GF(3) completely to hardware, and the issues that surrounded the migration. The original work [1] concentrated only on implementing the computation intensive *multmod* function of the GF3 algorithm in hardware.

2. BACKGROUND

This section will discuss the background information that is necessary for understanding this paper. This discussion includes Handel-C, Galois Fields and the previous work that was completed.

2.1 HANDEL-C

The hardware implementation for this work was implemented in Handel-C [2]. Handel-C is a high level hardware description language that bears much resemblance to the ANSI C programming language. While Handel-C is very similar to ANSI C in many respects, there are some major

differences between the two languages. Handel-C does not support the entire ANSI C specification. One of the more important features removed from Handel-C is support for runtime recursion. Handel-C, along with support for a subset of the ANSI C specification, includes extra support for hardware descriptions. Included in this extended support are constructs for input and output, communications, and control flow constructs for controlling the parallelism of a design. Parallelism in a Handel-C program is defined by using the `par{}` and `seq{}` statement blocks. Sequential instructions wrapped in a `par{}` statement will be executed parallelly, while statements wrapped in a `seq{}` statement will be forced to execute sequentially. Example 1 shows `par` and `seq` statements in a simple Handel-C design.

```
int 8 a,b,c,d,e,f,g,h;
a = 1; b = 2; c = 3; d = 4;
par {
        d = a + b;
        e = c + d;
}
seq {
       f = d+e;
       g = d*e;
}
```

Example 1: Example of `par` and `seq` Statements

The absence of runtime recursion support in Handel-C proved to be one of the more challenging aspects of this work. In most cases recursive algorithms can be easily converted to a non-recursive, loop based algorithm. This would prove to be problematic during the course of this work as several of the recursive functions written in the C algorithm proved to be resistant to conversion loops.

2.2 GALOIS FIELDS AND THE ALGORITHM

A Galois Field is a finite order denoted by GF(p) where p is a prime or a power of primes [3]. A Galois Field of order p has only p elements, 0 though p-1. The focus of the algorithm implemented for this paper is Galois Fields of the order GF(3). These fields are of interest due to their application in pairing based cryptographic systems [4].

The C algorithm discussed in this paper describes the problem of enumerating all of the primitive and irreducible polynomials of a given order [5]. Irreducible polynomials are polynomials such that p(x) in F(x) is called irreducible over F if it is non-constant and cannot be represented as the product of two or more non-constant polynomials from F(x) [3]. A primitive polynomial is a polynomial such that F(X), with coefficients in GF(p) = Z/pZ, is a primitive polynomial if it has a root α in GF(pm) such that $\{0, 1, \alpha, \alpha^2, \alpha^3, \ldots, \alpha^{p^m-2}\}$ is the entire field GF(p^m), and moreover, $F(X)$ is the smallest degree polynomial having α as root [3].

The C algorithm consists of a number of functions that will now be detailed. Where applicable functions that are recursive are noted.

Add: Adds two polynomials under GF(3).
Subtract: Subtracts two polynomials under GF(3).
Mod: Takes the modulus of two polynomials under GF(3).
GCD: Find the greatest common divisor of two polynomials under GF(3) (recursive).
Multmod: Multiplies two polynomials under mod p.
Powmod: Finds the result of one polynomial raise to the power of another polynomial under GF(3).
Minpoly: Finds the minimum polynomial given a necklace.
Gen: Controls execution of the algorithm (recursive).

2.2 THE CO-DESIGNED SOLUTION

The previous implementation of the C algorithm did not attempt to migrate the entire software algorithm into a hardware system. Instead it was decided to explore a co-designed approach [1] where only a portion of the software would be translated into a hardware design and this hardware module would be called from the software running on a general purpose CPU.

After profiling the C algorithm it was decided that the *multmod* function was the most computationally intensive function found within the software, thus *multmod* was chosen as the function to be implemented in hardware.

The hardware implementation of the *multmod* algorithm was implemented in Verilog and was targeted to an Amirix AP1000 [6] development board. This development board was chosen as the target platform because of its on-chip PowerPC processor that is directly connected to the FPGA fabric. This feature allowed the software to be executed on a platform that is more tightly coupled with the FPGA and removed the need to create a PCI bus driver for the work. Figure 1 shows an overview of the co-designed system and Table 1 shows the benchmarking results for this implementation.

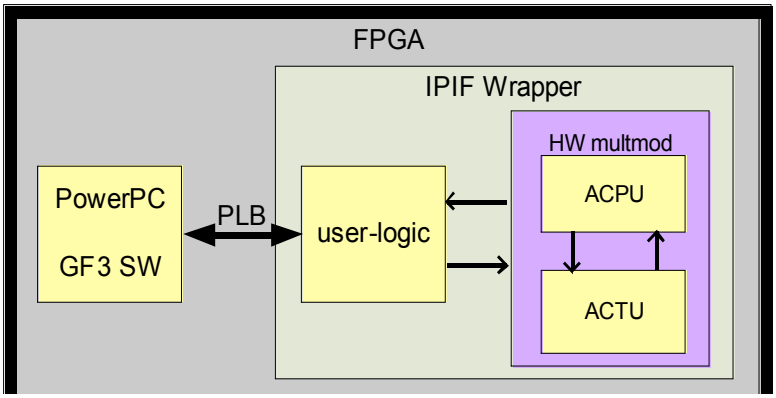


Figure 1: Co-Designed System Overview[1]

While a performance increase was realized by moving to the co-designed system, it was found that several factors severely limited the overall performance of the system. The slow speed of the embedded processor running the software portion of the system was one issue that arose. The 200mhz clock speed of this processor simply was not fast enough to hold pace with the faster general purpose processors that would normally run the full software implementation [1].

Also a major problem, more so than the slow clock rate of the embedded processor running the software portion of the system, is the communications between the hardware and the software system.

Communications prove to be the Achilles heel of this work, as well as many other co-design works [7]. The amount of data communications that is necessary between the hardware and the software is so great that it limits the maximum throughput of the system, which has a huge impact on performance.

The only solution to this problem is to move the entire system into hardware, completely eliminating the communication channels. This will allow the system to operate at full speed, only having to access communication channels when retrieving jobs and reporting results.

Col. 1	Col. 2	Col. 3	Col. 4	Col. 5	Col. 6
Degree	**Pentium @1.8GHz runtime (sec)**	**Altera FPGA@66MHz runtime (sec)**	**%**	**Xilinx FPGA@80MHz runtime (sec)**	**%**
2	0.00004	0.00002	*37.6*	0.00001	*24.8*
3	0.00018	0.00009	*46.6*	0.000056	*30.7*
4	0.00076	0.00034	*44.2*	0.000228	*30.0*
5	0.00298	0.00145	*48.6*	0.001005	*33.7*
6	0.01495	0.00519	*34.7*	0.003664	*24.5*
7	0.04043	0.02005	*49.6*	0.01439	*35.6*
8	0.14300	0.07123	*49.8*	0.051788	*36.2*
9	0.54600	0.25595	*46.9*	0.188174	*34.5*
10	1.89800	0.89412	*47.1*	0.663513	*35.0*
11	6.24000	3.08083	*49.4*	2.302203	*36.9*
12	22.30800	10.58020	*47.4*	7.963267	*35.7*
13	74.78900	35.12896	*47.0*	26.53804	*35.5*
14	263.53600	120.09697	*45.6*	91.323996	*34.7*
15	888.30300	400.77343	*45.1*	306.075094	*34.5*
16	2985.50200	1343.56091	*45.0*	1049.41517	*35.9*
17	10192.85900	4424.87400	*43.4*	n/a	n/a
18	32658.34090	14642.10675	*44.8*	n/a	n/a

Table 1: Co-designed Performance Results [1]

3. THE HARDWARE SOLUTION

In order to alleviate the performance degradation caused by communications between the hardware and software in the co-designed system, as well as the low performance of the general purpose processor, a full implementation was created in hardware. This implementation was written in Handel-C which allowed the hardware implementation to very closely mimic the software algorithm wherever possible.

Much of the ANSI C code that was created for the algorithm was capable of being directly translated into Handel-C. The code that was directly translated required only minimal modification to make it compatible with the Handel-C language. Some of these changes included re-definition of storage elements such as arrays to use static sizes instead of being dynamically allocated. Another trivial modification that was required in several places was the un-nesting of function calls. Handel-C does not support the usage of nested function calls of the form `foo( bar(x,y),z)`. This

necessitated rewriting some C code to call these functions sequentially using temporary variables to store the return value of the nested function call.

Once the code was converted to Handel-C syntax all that remained was removing the recursion that exists in several of the functions in the software. The functions that required modification to remove recursion were the Gen and GCD functions. Both functions were translated to their loop based variants. Example 1 shows how the recursive function definition for the GCD was transformed into a loop.

```
Poly_GF3 gcd(Poly_GF3 a, Poly_GF3 b){
  if(!b.top && !b.bot) return a;
  return gcd(b, mod(a, b));
}
```

Example 1 (a): Recursive GCD Definition

Once the recursion was removed from the software functions they were implemented in Handel-C. Following the implementation in Handel-C, each function required verification to ensure that the hardware versions were equivalent to their software counterparts.

```
Poly_GF3 gcdx(Poly_GF3 a, Poly_GF3 b )
{
  Poly_GF3 c,zero;

  zero = {0,0};
  while (a.top || a.bot)
  {
     c = a;
     modx(b,a);
     a = modxResult;
     b = c;
  }
  return b;
}
```

Example 1 (b): Non Recursive GCD Definition

3.2 HARDWARE VERIFICATION

In order to verify that the hardware functions, especially the functions that were transformed from recursive to non-recursive, behaved as intended it was necessary to perform some verification tests. Test cases included boundary cases as well as a large number of randomly generated inputs to the functions.

Verification of the transformed recursive functions was performed in two stages. In the first stage, the non-recursive algorithm was tested as a software algorithm. Test cases were run against both the recursive and non-recursive versions of the functions and their return values were compared. Following running the test cases on both the recursive and non-recursive functions it was deemed that the recursive and non-recursive functions were both functionally equivalent and so passed verification.

Verification of the Handel-C hardware code was slightly more involved than testing software code against software code. The Handel-C hardware code was again tested using the same set of test cases used for testing the recursive functions. These test cases were first ran in the software version of the system, recording the results for each test. The same tests were then performed on each hardware function individually, running the hardware in a simulation environment. The results were also recorded and compared to those produced by the software for the same tests.

Following verification of the hardware definition it was deemed that the hardware definition is equivalent to the software algorithm so the work could proceed to benchmarking.

4. Benchmarking

In order to benchmark the hardware design of the GF(3) algorithm, it was necessary to synthesize the hardware definition to produce a hardware programming file. It was decided that the hardware would not be programmed onto a physical device for testing, but tests would be performed in a simulation environment in order to facilitate the gathering of statistics.

The Handel-C definition was first compiled using the Agility Handel-C compiler to produce both an executable simulation file as well as a synthesizable VHDL description file. The execution simulation kernel was used to gather timing results for the hardware system and the VHDL description file was used to gather resource usage and clock speed statistics. Resource usage and clock speed statistics were gathered by synthesizing the VHDL specification in Xilinx ISE targeting a Virtex II FPGA (XC2VP100). This FPGA is the same device used for gathering the results for the co-designed GF(3) algorithm. The results in table 2 show the resource usage and clock frequency for the design.

Clock Speed	Slices	Flip flops
68.523	23952	14579

Table 2: Resource Usage and Clock Frequency

Runtimes for the hardware were gathered by running the simulation kernel on different degrees ranging from 3 to 12. Cycle statistics were gathered for each run, and using the clock rate gathered from the Xilinx synthesis tool a run time was calculated. These run times are compared to the runtimes of the software in Table 3. Software run times were gathered on a 2.8 Ghz Pentium 4 with 2 Gb of RAM.

N	Cycles	HW Time (Seconds)	SW Time (Seconds)
3	15158	0.000212	0.0127
6	899241	0.0131	0.0178
8	13052272	0.1905	0.1347
10	170959343	2.4949	1.5062
12	2072543280	30.2495	21.8374

Table 3: Runtime Comparison

On inspection of the results, it can be clearly seen that the hardware version of the algorithm, in its current form, does not surpass the performance of the software algorithm. While the hardware algorithm does not perform better than the software, the performance gap between the two is negligible when taking into account the speed grade difference between the hardware running at 68.523 Mhz and the software running on a 2.8 Ghz processor.

Taking this into account it was decided to attempt to improve the hardware design further by attempting to optimize the design for a hardware environment. Until this point the software had been converted to a hardware definition almost verbatim, ignoring any of the traditional hardware specific features such as parallelism.

5. Optimization

The optimization that was chosen for this design was the addition of parallelism to the design. The software design did not take into account any of the areas of parallelism that might lead to greater performance for the hardware system. For the purpose of this work, only simple optimizations were attempted. Individual statements that were capable of parallel execution were grouped into parallel blocks using the Handel-C `par` construct.

The parallel blocks were identified using a combination of both an automated parallelism detection tool [12] as well as manual optimization. This tool allows for the automatic identification of code that can potentially be executed in parallel. Currently the tool does not modify the Handel-C source directly and requires intervention from the designer to take advantage of code that is identified as parallel. The automated tool found a large portion of the available parallel blocks, and then manual code inspection was used to find more parallel blocks that the tool was unable to identify.

After optimization of the hardware algorithm 17 `par` blocks of two or more sequential statements were identified. Parallel execution statements (`par{}`) were added to the design and the design was recompiled, again producing both a simulation kernel and a VHDL definition file for hardware synthesis. Table 4 shows the synthesis results gathered from the Xilinx ISE, again targeting the Virtex II FPGA (XC2VP100).

Clock Speed	Slices	Flip flops
68.813	23348	14245

Table 4: Resource Usage and Clock Frequency

Using the clock speed from Table 4 and the statistics gathered from the simulation kernel the runtime statistics for the hardware algorithm can be calculated. Table 5 shows the new runtimes for

the parallel hardware design. Also shown in Table 5 is the percentage reduction of clock cycles between the original non-parallel design and the parallel design.

N	Cycles	Percent Reduction	HW Time (Secs)	SW Time (Secs)
3	8621	43.1%	0.000125	0.0127
6	548189	39.0%	0.0079	0.0178
8	8089562	38.0%	0.1176	0.1347
10	106849548	37.5%	1.5527	1.7392
12	1352768511	34.7%	19.6586	21.8374

Table 5: Parallel Runtime Comparison

Table 4 shows that a small increase, 0.290 Mhz, in clock speed was realized when moving from the non-parallel to the parallel design. The number of slices and flip flops utilized by the design was also reduced slightly. Figure 2 shows a comparison of the parallel and non-parallel hardware against the software implementation.

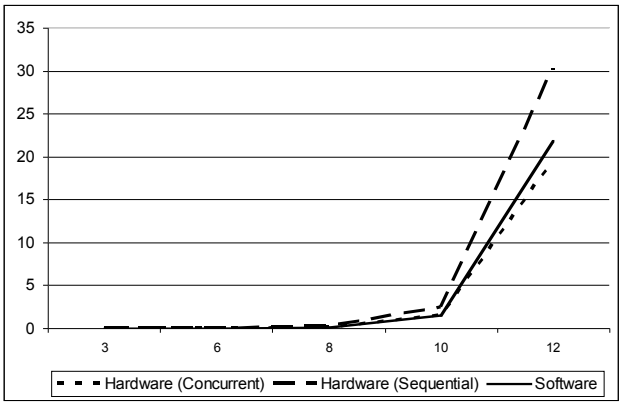


Figure 2: Results Comparison

It can be seen in Figure 2 that the parallel version of the hardware outperforms the software implementation of the algorithm at all data points gathered for this work. It also appears that the hardware will continue to outperform the software even when computing orders higher than 12. Figure 3 illustrates the trend in the percentage difference between the hardware and software algorithms. This figure clearly shows that the rate of convergence between the hardware and software run times is slowing and that the hardware will continue to outperform the software.

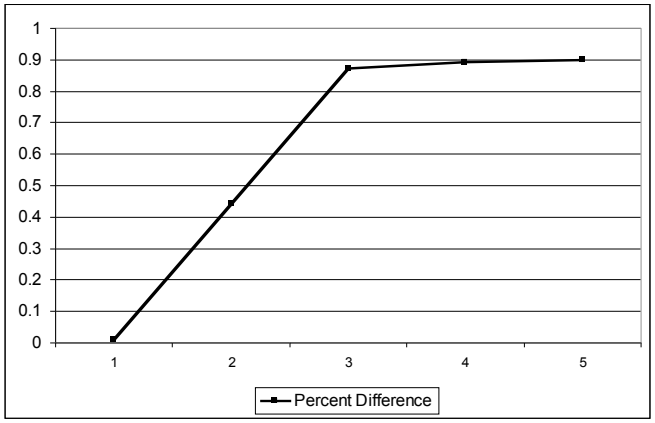


Figure 3: Execution Time Percentage Difference Between Hardware and Software

6. Conclusion

Based on the results gathered after optimizing the Handel-C design for the GF(3) primitive and irreducible polynomials algorithm it can be said that this work is a success. The entire algorithm was implemented in hardware and verified to function correctly. The results found in Section 5 highlight the performance of the hardware system, which outperforms the software on all test points up to order 12. It also appears that, based on Figure 2, the software will continue to outperform the hardware on higher orders.

7. Future Work

While the work can be considered a success, there is still much work to be done to further improve the performance of the system. At present only simple parallelism has been identified in the system. While parallelism between individual statements in a Handel-C program can greatly increase performance, there can be even greater performance gains from exploiting loop based parallelism or parallelism between different functional units.

Another optimization that may greatly benefit this work is the identification and implementation of a pipelined data path. A pipelined data path may increase the throughput of the algorithm by increasing the amount of work that is done per clock cycle by breaking the algorithm down into functional units that can operate in parallel much like an assembly line.

References

[1] K. Kent, B. Iaderoza, M. Serra. Codesign of a Computationally Intensive Problem in GF(3), International Workshop on Rapid System Prototyping 2006.

[2] Agility Design Solutions, Handel-C Reference Manual, Website: www.agilityds.com. Accessed: May 15, 2008

[3] G. Birkhoff, S. Mac Lane. A Survey of Modern Algebra, 5th ed. New York: Macmillan, 1996.

[4] D. Page and N. P. Smart, "Hardware Implementation of Finite Fields of Characteristic Three". Proc. of the CHES 2002, 2002.

[5] G. Lee, F. Ruskey, Listing all Irreducible and Primitive Polynomials in GF(3),. Technical Report (UVic), 2006. Unpublished.

[6] AP1000 FPGA Development Board User Guide. User Guide Manual Version 2. AMIRIX Systems Inc, Halifax, Nova Scotia, Canada. 2005.
[7] M. Moazeni, A. Vahdatpour, K. Gururaj, and M. Sarrafzadeh, Communication Bottleneck in Hardware-Software Partitioning. In Proceedings of the 16th international ACM/SIGDA Symposium on Field Programmable Gate Arrays, 2008.
[8] R. Andraka, A Survey of CORDIC Algorithms for FPGA based Computers, 1998 ACM/SIGDA 6th Int. Symp. Field Programmable Gate Arrays.
[9] M. Mylona, D. Holding, and K. Blow, DES Developed in Handel-C, London Communications Symposium, 2002.
[10] Serra, M., and K. Kent, Using FPGAs to Solve the Hamiltonian Cycle Problem, ISCAS, 2003.
[11] Tobias G. Noll, Application Specific eFPGAs for SoC Platforms, 2005 IEEE VLSI-TSA Int. Symposium on VLSI Design, Automation and Test. April 2005.
[12] Joseph C. Libby, Kenneth B. Kent, Automatic Identification of Concurrency in Handel-C, International Symposium on Digital Systems Design, 2008.

Ontology-based Unit Test-case Generation

Valeh H. Nasser, Weichang Du, Dawn MacIsaac
Faculty of Computer Science, University of New Brunswick
Fredericton, NB, Canada
{valeh.h, wdu, dmac}@unb.ca

Abstract

In software unit testing, to identify test objectives, various coverage adequacy criteria are suggested. This paper proposes to use reasoning on ontologies to generate unit test objectives. Ontologies are used for specification of test-oracles and a test-suite, and rules are used for specification of coverage criteria. Knowledge externalization, in contrast to hardcoding in algorithms, enables test experts to specify coverage criteria and to enrich test oracles with different pieces of knowledge. Afterwards, the generated test objectives need to be implemented. An architecture for the system and implementation technologies which are used are described.

1. Introduction

Unit testing is testing the smallest unit of a system under test. It is important because it reduces the cost of software testing, by discovering errors before they affect a larger portion of the system. While testing reduces costs by elevating the quality of the unit under test, the testing activity is costly itself. Hence, the quality of a test suite has a direct relation to the number of errors it discovers, and an inverse relation with its cost. As a result, specification of an optimum test suite is crucial. To specify how much testing is enough and what needs to be tested, coverage adequacy criteria are used [22].

Test cases can be generated from various software artifacts, namely: code, design, and requirements [17]. Being at different levels of abstraction, the knowledge that these artifacts provide for test-case generation is different. This knowledge is used by the test coverage adequacy criteria to identify what needs to be tested.

While abstraction is used to concentrate on important aspects of the system at hand, poor abstraction can be a barrier for generating good tests [4]. Poor abstraction of the test-oracle could remove the knowledge that helps identification of risky test-cases. Benz [4] demonstrates how abstraction of defect-prone aspects of software can enhance test-case generation by defining system-specific coverage criteria.

One of the common models that is widely used in unit testing is the UML State Machine. Many methods that use UML state-machines for test-case generation, are based on some coverage adequacy criteria. Coverage adequacy criteria rules are the explicit specification for test selection and specify what needs to be observed [22]. Zhu et al. [22] categorize coverage adequacy criteria as structural testing (such as All Transition coverage, All Transition Pair coverage, Full Predicate coverage [16], Faulty Transition Pair coverage [3], All Content Dependence Relationships [21], Session based and, 2-Way criteria[18]), fault-based testing (such as plannable test selection criteria [18]) and error based testing (such as Boundary Testing [13]).

Another aspect of test case generation is the method which is used for specification of what needs to be tested. Some of the automated test case generation methods use explicit specification of test-cases [11]; some others use rules which are implicit in the algorithms to generate test cases [16]. Another approach for specifying coverage criteria is to provide a language for defining rules for generating test cases [7]. GOTCHA [7] uses a notation to define partitions of states and transitions to specify what needs to be covered and what states, sub-paths, and transitions would be ignored. The drawback of GOTCHA is that the state-space needs to be finite; as a result, for instance, modeling an unbounded-buffer is not straight-forward.

Besides the specification of the coverage criteria, another challenge to state-machine based testing methods is the generation of test-cases. In the state machine based test case generation, the test cases are paths from the start state to a final state. There are several approaches to generation of the test cases. One approach is to use graph traversal algorithms [16, 3, 7]. Another approach is using model-checking tools for test-case generation [20]. With this approach it is asserted that there is no path with the required specification in the model. The model checker tries to find the required path and returns it as output. A third approach is using AI planners to generate test-cases [18]. AI Planners are used to generate paths to reach identified goals.

The objective of the present work is to improve unit test case generation by generating test objectives from modifiable test oracles and coverage adequacy criteria specification. Then, for each test objective, a test case needs to be generated. Different systems have different erroneous aspects that need to be modeled and coverage criteria based on these abstractions need to be defined and used [4]. A modifiable test oracle that allows specification of arbitrary test cases, and can be extended with implementation knowledge, invariants on model elements, distinguished states [19], and allows adding knowledge about erroneous aspect of the system, is required to increase the control of the test-expert on the test-case generation. In this paper, an ontology and rule-based method for unit test case generation is proposed.

The rest of this paper is organized as follows: Section 2 is an overview of the system, Section 3 describes implementation, Section 4 concludes the paper and envisions future works.

2. Ontology-based Unit Test Generation

In this work, knowledge engineering technologies are exploited to externalize the knowledge that is used in the process of the test-case generation. This knowledge can be modified and extended for specifying different aspects of test oracles such as implementation knowledge, error-prone aspects, and other invariants which are needed for generation of arbitrary test-cases. In this regard, ontologies are proposed to be used for specification of the test-oracle and rules for

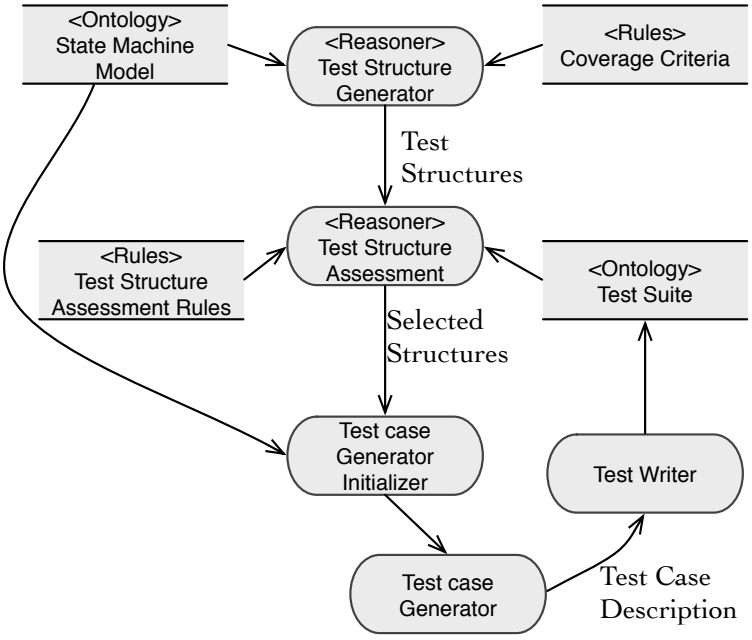


Figure 1. System architecture

specification of the coverage adequacy criteria. The ontology and rules can be used to implement several existing structural coverage criteria, and modifications to them (for instance on their domain). Furthermore, they provide extendibility for supporting other user-defined coverage criteria. With this specification and use of reasoning, test-case structures are specified. Then, test-cases are generated using a test-case generation method such as AI planners. Furthermore, the proposed system avoids generating redundant test cases for a test structure by specifying the test suite in an ontology and reasoning on it to determine whether a test-case with a specific test structure already exists in the test-suite or not. To the knowledge of the authors, the use of ontology and reasoning for test-objective generation has not been explored yet.

Figure 1 shows the architecture of the system. First, an ontology representing the state machine model and rules specifying the coverage criteria are provided as inputs to the Test Structure Generator process, which uses reasoning to generate test structures. Coverage criteria rules are in the following form:

```
test structure :- test structure selection criteria
```

The test structure selection criteria specify a condition that should hold on some model elements for them to be a part of structure of a test case. Next, the generated test structures are processed by the Test Structure Assessment process which uses reasoning to assess whether a given test-structure already exists in the test-suite or not. If it does not exist, the test structure is accepted, otherwise it is discarded. Other inputs into this process are the test-suite ontology and assessment rules. If the given test structure is accepted, then a test case is generated for it and added to the test suite ontology. Then, the test structure assessment reasoner continues to select another test-structure.

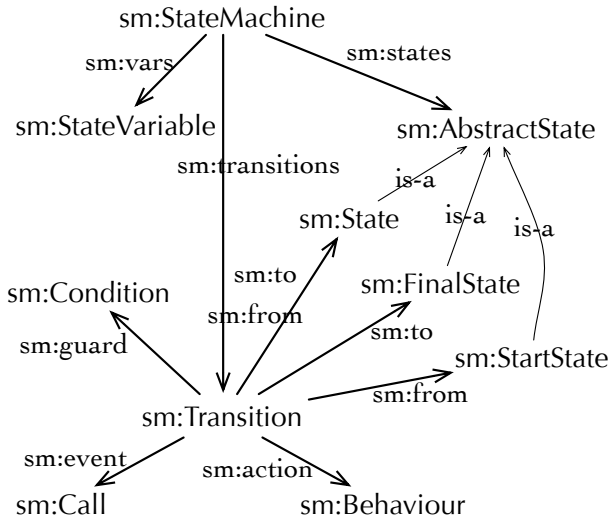


Figure 2. Part of the state machine T-Box

An ontology and rules on the ontology form a flexible mechanism for specification of various coverage criteria. The system empowers the test expert to specify coverage criteria; it enables the test expert to use knowledge about the system to control the size of the test-suite by specifying what needs to be tested. It empowers the user to add different pieces of knowledge to the ontology model of the unit under test and to use this knowledge in the specification of coverage criteria.

3. Implementation

The state machine model and test suite ontologies are represented in OWL [2]. The state machine model can be converted from XMI [15] to an ontology-based representation; a T-Box ontology specifies different concepts and relations in a state machine. An A-Box ontology specifies a state machine instance by importing the T-Box ontology and instantiating the elements in it. The Ontology Definition Metamodel (ODM), which is adopted by the OMG, has a section that describes the UML 2.0 metamodel in OWL. A rough implementation of the ODM is found in [14]. The ODM is not finalized yet and a prototype ontology is used for the purpose of this work. Some parts of the T-Box of the state machine model and the test-suite ontologies are visualized in Figures 2 and 3 respectively.

The coverage adequacy criteria and test structure assessment rules are written in Positional-Slotted Language, POSL [5]. For reasoning, the ontologies are mapped to POSL using the mappings suggested by Grosof et al. [8]. OO jDrew [1] is then used for reasoning. For instance, coverage criteria rule for All Transition Pair Coverage (ATP) [16] is shown below:

```
ATP: coverage([immediate],[?t2,?t1]) :- transition(?t1),
transition(?t2), notEqual(?t1,?t2), from(?t1,?state), to(?t2,?state).
```

The head of the rule specifies a test structure, which is implemented using two lists: the list of

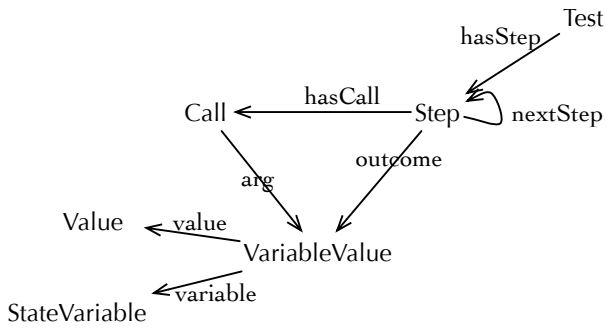


Figure 3. Part of the test suite T-Box

test structure predicates and the list of their arguments. The body of the rule denotes the conditions, which should hold on the model elements for them to be a part of the test structure.

The All Transition Pair coverage criteria only depends on the specification of the structure of a state machine; that is the specification of the standard UML state machine elements. Additional knowledge can be added to the UML state machine ontology, and be referred to in coverage criteria rules.

After a test structure is generated, it is assessed whether a test case conforming the structure already exists in the generated test suite. For this purpose, the partially generated test suite is represented in an ontology, and for every generated test structure, an structure assessment rule is generated. Part of the test suite ontology T-Box is visualized in Figure 3. The test suite ontology is converted to POSL and OO jDREW is used for reasoning. An example of a test structure assessment rule for the *immediate* test structure is given below:

```
Test Structure:  [immediate],[t1,t2]
Assessment Rule: exist([immediate],[t1,t2]):-
test(?t),  hasStep(?t,t1), hasStep(?t,t2), nextStep(t1,t2).
```

After the test structures are generated, we plan to use AI-Planning method for generation of test-cases from the generated test-structures. For each selected test structure, a planner named Metric-FF [10] is run to generate test-cases. It uses the PDDL 2.1 [6] and supports numeric fluents. It may be difficult to utilize this approach with systems with more complex data structures, such as arrays, because of the low expressiveness power of the PDDL. Before the planner is run, the PDDL domain and problem descriptions need to be generated. The PDDL Domain and Problem objects are initialized using the model ontology. For the domain description, a state is mapped to a PDDL type, a transition is mapped to an action, and a transition guard is mapped to an actions precondition. An active predicate indicates the active state and is added to the action's precondition and effect. Initially the start state is active. A transition action is mapped to the effect of an action to change the value of the state variables. The state varibles are implemented using additional fluents and predicates. In the problem description, an object for each PDDL type which correponds to the states is defined. The initial and goal conditions map to active predicate of first state and final state. Then for each test-structure, the PDDL Domain and Problem objects are cloned, and predicates

are added to them to implement the test structure. Next, the domain and problem descriptions are edited and written to a file; then, Metric-FF is executed, and the generated plan is written back to the ontology illustrated in Figure 3.

4. Concluding Remarks

In this work, it is proposed that rules on the model ontology can serve as a notation for specifying test coverage criteria and reasoning can be used for generating test objectives. Several components of the system are highly modifiable: the test case generation algorithm can be changed; the coverage adequacy criteria and assessment rules can be modified. The knowledge that is used by the coverage criteria can be added to the model ontology. The tester can control and adapt the test suite based on their knowledge about the system. This system can be extended using reverse engineering to populate the interesting implementation knowledge such as definition use relationships from the code automatically. The high level of modifiability of the system is due to externalization of knowledge and use of general purpose reasoning algorithms.

The test-case generation needs to be implemented using an AI planner to use mutation analysis [12] for testing the effectiveness of the test suite. Because of limitations of the modelling power of AI planners in modeling complex state variables and and changes to them, the domain of the systems that can be tested is limited to the ones that only have integers and booleans. Model-checkers such as [9] are more mature in simulating the behaviour of the system and needs to be exploited for practical use of the system.

References

[1] M. Ball. OO jDREW: Design and Implementation of a Reasoning Engine for the Semantic Web. Technical report, Technical report, Faculty of Computer Science, University of New Brunswick, 2005.

[2] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. McGuinness, P. Patel-Schneider, L. Stein, et al. OWL Web Ontology Language Reference. *W3C Recommendation*, 10:2006–01, 2004.

[3] F. Belli and A. Hollmann. Test generation and minimization with" basic" statecharts. In *Proceedings of the 2008 ACM symposium on Applied computing*, pages 718–723. ACM New York, NY, USA, 2008.

[4] S. Benz. Combining test case generation for component and integration testing. In *Proceedings of the 3rd international workshop on Advances in model-based testing*, pages 23–33. ACM Press New York, NY, USA, 2007.

[5] H. Boley. POSL: An Integrated Positional-Slotted Language for Semantic Web Knowledge. http://www.ruleml.org/submission/ruleml-shortation.html, 2004.

[6] M. Fox and D. Long. PDDL2. 1: An extension to PDDL for expressing temporal planning domains. *Journal of Artificial Intelligence Research*, 20(2003):61–124, 2003.

[7] G. Friedman, A. Hartman, K. Nagin, and T. Shiran. Projected state machine coverage for software testing. In *Proceedings of the 2002 ACM SIGSOFT international symposium on Software testing and analysis*, pages 134–143. ACM New York, NY, USA, 2002.

[8] B. N. Grosof, I. Horrocks, R. Volz, and S. Decker. Description logic programs: combining logic programs with description logic. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 48–57. ACM, 2003.

[9] G. Holzmann. *The Spin Model Checker: Primer and Reference Manual*. Addison-Wesley Professional, 2004.

[10] J. Homann. The Metric-FF Planning System: Translating "Ignoring Delete Lists" to Numeric State Variables. *Journal of Artificial Intelligence Research*, 20:291–341, 2003.

[11] A. Howe, A. Mayrhauser, and R. Mraz. Test Case Generation as an AI Planning Problem. *Automated Software Engineering*, 4(1):77–106, 1997.

[12] S. Kim, J. Clark, and J. McDermid. The Rigorous Generation of Java Mutation Operators Using HAZOP. In *Proceedings of the 12th International Conference on Software and Systems Engineering and their Applications*, 1999.

[13] N. Kosmatov, B. Legeard, F. Peureux, and M. Utting. Boundary Coverage Criteria for Test Generation from Formal Models. In *Proceedings of the 15th International Symposium on Software Reliability Engineering*, pages 139–150. IEEE Computer Society Washington, DC, USA, 2004.

[14] E. Lehtihet. http://www.tssg.org/public/ontologies/omg/uml/2004/UML2-Super-MDL-041007.owl, May 2005.

[15] Object Management Group. XML Metadata Interchange (XMI) specification. http://www.omg.org/technology/documents/formal/xmi.htm, 2007.

[16] J. Offutt and A. Abdurazik. Generating tests from UML specifications. In *UML'99 - The Unified Modeling Language. Beyond the Standard. Second International Conference, Fort Collins, CO, USA*, volume 1723, pages 416–429. Springer, 1999.

[17] T. Ostrand and M. Balcer. The category-partition method for specifying and generating fuctional tests. *Communications of the ACM*, 31(6):676–686, 1988.

[18] A. Paradkar. Plannable Test Selection Criteria for FSMs Extracted From Operational Specifications. In *Proceedings of the 15th International Symposium on Software Reliability Engineering*, pages 173–184, 2004.

[19] A. Paradkar. A quest for appropriate software fault models: Case studies on fault detection effectiveness of model-based test generation techniques. *Information and Software Technology*, 48(10):949–959, 2006.

[20] S. Rayadurgam and M. Heimdahl. Coverage based test-case generation using model checkers. In *Engineering of Computer Based Systems, 2001. ECBS 2001. Proceedings. Eighth Annual IEEE International Conference and Workshop on the*, pages 83–91, 2001.

[21] Y. Wu, M. Chen, and J. Offutt. UML-Based Integration Testing for Component-Based Software. In *Cots-Based Software Systems: Second International Conference, ICCBSS 2003, Ottawa, Canada*, pages 251–260. Springer, 2003.

[22] H. Zhu, P. Hall, and J. May. Software unit test coverage and adequacy. *ACM Computing Surveys (CSUR)*, 29(4):366–427, 1997.

Knowledge Base Validation under Closed-World Semantics

Cheng Lu
Faculty of Computer Science

Abstract

It is now possible and in some cases desirable to use a Knowledge Base (KB) to store business data. However, the common validation operation applied to KB does not always perform as expected from the business perspective. Knowledge Base typically is supposed to represent an "open-world". For the traditional database, the data domain is always "closed". This difference is often referred to as "open-world" vs. "closed-world". Most common Description Logic (DL) reasoners follow the open-world standard when performing reasoning tasks on KBs. The results from these reasoning tasks do not always satisfy the users who view the KB with a database perspective which typically is closed-world. In this paper, we propose an approach validating a KB under the closed-world semantics. We design and implement a DL reasoner prototype which is capable of dealing with both open-world and closed-world reasoning tasks. Reasoning with a KB that is partially closed using the 'K' operator is also discussed in this paper. Traditional database users will have a flexible way to express that some parts of the KB are open and some are closed by using K-operator reasoning services.

1 Introduction

Validation is a very important part of database management. For the integrity of a business database, invalid data needs to be either removed from the database or updated with necessary information. The most common database, the relational database, has a method to add constraints for a single data attribute or an entire relation tuple. But a relational database schema does not have the capability to describe some high-level abstract property constraints for business data. A Knowledge Base (KB) overcomes this shortcoming by allowing data entity constraints to be associated directly with the data.

Using a business KB to replace the business database will make it easier to add validation constraints for data. Using a business KB also brings benefit for database developers because they can use one general DL ontology reasoner to validate any business KB instead of writing a validation program for each specific business database. It is also not an easy job for developers to integrate two databases constructed in different languages. By using KB technology, they can describe the business information in the ontology form and make further integration easier.

However, the KB represented in the ontology form brings in some new problems in terms of validation. We illustrate this with two examples. One shows how ordinary integrity constraints can be violated and the other shows how number constraints, known as cardinality constraints, can be violated.

For the first example, assume in a 'Bank Account' Knowledge Base, we have this information about accounts and customers described as follows:

- Account is a subset of things that have an owner who is a Customer.
- '#326974' is an Account ID.
- '#275482' is an Account ID.
- 'ID41981' is a Customer ID.
- the owner of Account '#326974' is Customer 'ID41981'.

This KB information would be written in DL form as:

$Account \sqsubseteq \exists hasOwner.Customer$
$Account(\#326974)$
$Account(\#275482)$
$Customer(\text{ID41981})$
$hasOwner(\#326974, \text{ID41981})$

The example above shows that this KB is not consistent with what we have in a traditional database system. For individual '#275482', it has been declared as member of class 'Account', but we cannot find its corresponding 'hasOwner' property statement in the KB. It seems that this KB is not complete since necessary information is missing. But surprisingly, typical DL reasoner will not inform the user that some necessary information is missing. This is because this KB is not inconsistent under the open-world view. More information could be revealed later that would satisfy the constraint. Thus the KB is not unsatisfiable; it is consistent.

Here is another example of a 'Car Registration' Knowledge Base, the information about the car related documents are:

- Registration Document is a subset of things which have at least 2 associated documents.
- 'R13821' is a Registration Document.
- 'DL3224' is a Driver License Document.
- 'CI45772' is a Car Insurance Document.
- 'R13821' has Driver License Document 'DL3224' associated with it.

This KB information would be written in DL form as:

$RegistrationDoc \sqsubseteq \geq 2\ hasAssociateDoc$
$RegistrationDoc(\text{R13821})$
$DriverLicenceDoc(\text{DL3224})$
$CarInsureDoc(\text{CI45772})$

hasAssociateDoc(R13821, DL3224)

This example is very similar to the first example, we know that registration document 'R13821' is associated with driver license 'DL3224', but we are not sure if 'R13821' associated with car insurance 'CI45772'. Although we think there is lack of one more 'associate' property statement for registration document 'R13821', once again, the DL reasoner will not complain about this as inconsistency in the KB. It seems that the min cardinality constraint defined for class 'RegistrationDoc' has not been validated at all.

The two problems arise because of the open characteristic of KB. Some information is not explicitly written in the KB, but this does not necessarily mean it does not exist. And this piece of information about the KB may be revealed in future. Thus the KB as a whole is still consistent under the open-world view of KB theory. Although KB provides us a convenient way to describe property constraints on entities, its open-world characteristic make it more difficult to validate. For business application users, they usually prefer that the KB is totally closed during the validation process. And they want the min cardinality constraint and the existential quantifier constraint to be validated under the closed-world view. But current DL reasoners such as Pellet [7] and Fact++ [1] are designed for open-world reasoning. These reasoners use the open-world view to process the KB consistency check.

This common problem for current DL reasoners motivates us to research how to validate and reason with a KB under closed-world semantics. We build a DL reasoner prototype which can query the KB and draw conclusions based on the closed-world view. Moreover, we extend the reasoning function to focus on reasoning with concrete data[1] explicitly stored in the KB, not the inferred data represented by the DL quantifier constraints. We also use the K-operator to extend the query language so the user can express where open-world view and where closed-world view is to be applied for a KB. These improvements would satisfy the needs for many business KB applications.

The rest of the paper is structured as follows. In section 2, we explain how current instantiation check service work under the open-world semantics, and then introduce how we adapt it into a closed-world reasoning service. Section 3 describes current progress of our research. Section 4 shows the related work done by other research groups on the K-operator syntax and semantics, and section 5 concludes.

2 Methodology

The most simple and basic validation function for a KB, is the instantiation query. It checks whether a specific instance, which is mentioned in the KB domain, instantiates a specific concept description or not. For each instantiation query, there is one situation for which open-world reasoning would answer "unknown". This is caused by the incomplete information in the KB ABox. The ABox information of a KB does not describe a particular state of that KB. It actually constrains the possible worlds that the KB describes. So there exist infinite possible worlds(models) which can be interpreted from the KB ABox. We customize the KB in two steps to achieve the

[1] concrete data represents concrete instantiation assertions in "a∈A" form and role assertions in "(a, b):R" form.

closed-world instantiation query. In the first step, we eliminate all the "unknown" answers by assuming the information is complete in the KB. If the reasoner cannot conclude "yes" or "no" in a certain situation, then we make the reasoner answer 'no'. After this setup, a KB's response to a DL instantiation query based on the corresponding KB information is shown in Table 1.

1	? a∈C
Yes	Every possible model of KB contains assertion a∈C
No	otherwise
2	? a∈∃R.C
Yes	Every possible model of KB contains either assertion set {(a, b):R, b∈C } or assertion a∈∃R.C
No	otherwise
3	? a∈∀R.C
Yes	Every possible model of KB contains a∈∀R.C
No	otherwise
4	? a∈≥nR
Yes	Every possible model of KB contains 1)at least n assertions of (a, b_i):R, and each b_i is distinct individual, or 2)assertion a∈≥nR
No	otherwise
5	? a∈≤nR
Yes	Every possible model of KB contains assertion a∈≤nR
No	otherwise

Table 1: Modified Instantiation Checking for a DL KB

However, the performance of this modified instantiation query does not totally satisfy the requirements for business KB validations. There are two major problems:

1) We can conclude from no.2 - no.5 of Table 1 that if a quantifier assertion appears in every possible world of a KB, plus it is consistent with KB ABox description under open-world semantics, then the instantiation query that is identical to this quantifier assertion would always receive the answer "Yes". The first step KB customization does not effectively validate the concrete data stored in a KB.

2)There is no way to receive 'yes' answers from instantiation queries corresponding to value restriction (?a∈∀R.C) and max cardinality(?a∈≤nR) based on concrete data in a KB under the open-world view. When generating models from a KB ABox, new facts can be revealed later as long as they do not contradict with facts that exist in the KB ABox. Therefore, we could always find some models interpreted from KB ABox where the value restriction or the max cardinality is violated by concrete facts unless the quantifier assertion itself exists in every model as 1) has described.

In order to solve the two problems above, we further customize the KB in the second step. We limit the number of possible models interpreted from the KB ABox by setting additional KB assumptions. Then we only need to consider about finite models interpreted from the KB ABox instead of infinite possible models for traditional open-world KB. The three additional KB

assumptions are the unique-name assumption, the domain-closure assumption, and the KB-closure assumption.

Unique-name assumption
The unique-name assumption is a basic assumption for database system. It says that two distinct constants (either atomic values or objects) necessarily designate two different objects in the universe. In Description Logic with the unique-name assumption, different names always refer to different objects in the KB domain. The Web Ontology Language (OWL) does not make this assumption as a default assumption, but provides a constructor to explicitly state that two individuals are different.

Domain-closure assumption
The database system is also based on another assumption called the domain-closure assumption. The domain-closure assumption suggests that "there are no other objects in the universe than those designated by constants of the database." [8].

KB-closure assumption
In order to solve the problem of "value restriction quantifier" validation, we need to introduce in a new assumption, which we call it the KB-closure assumption. The KB-closure assumption says in a KB model, we do not use the ALCN quantifiers including role existence quantifier assertion, value restriction quantifier assertion, min cardinality quantifier assertion and max cardinality quantifier assertion to represent knowledge any more. All the knowledge we know about a model must be represented by concrete instantiation assertions in "a∈A" form and role assertions "(a, b):R" form.

By applying the unique-name assumption, the domain-closure assumption and the KB-closure assumption to the KB, we now have a closed-world instantiation checking standard which meets the needs for common business KB validation purpose. The detail is shown in Table 2.

On the basis of the closed-world instantiation checking, we propose a theory for the closed-world KB consistency validation. It is the extension from the original consistency theory for the KB. The original standard for a consistent KB under open-world semantics is:

"There is at least one model that can be generated from the KB ABox."

Our proposed standard for a consistent KB under the closed-world semantics is:

"There is at least one model that can be generated from the KB ABox and all the quantifier assertions in this model are satisfied by concrete data that exist in the same model."

In this way, the validation within that model can get rid of the interference from uncertain data(e.g. a∈B⊔C). Users still have the freedom to express uncertain data in the KB. When the user provides only the concrete data and validation requirements, the KB ABox will generate at most one model. Our new theory for KB closed-world consistency validation in this situation will perform exactly like validation in a single-model database.

3 Results

The open-world reasoning services have been implemented for our reasoner prototype include concept satisfiability check, concept subsumption check, concept equivalence check, KB consistency

? a∈C	
Yes	Every model contains assertion a∈C
No	otherwise
? a∈∃R.C	
Yes	Every model contains assertion set {(a, b):R, b∈C }
No	otherwise
? a∈∀R.C	
Yes	For every (a, b):R role assertion in each model, there exists b∈C assertion
No	otherwise
? a∈≥nR	
Yes	Every model contains at least n assertions of (a, b_i):R, and each b_i is distinct individual
No	otherwise
? a∈≤nR	
Yes	Every model contains at most n assertions of (a, b_i):R, and each b_i is distinct individual
No	otherwise

Table 2: Closed-world Instantiation Checking for a Business KB

check and individual instantiation check. The closed-world reasoning services we have implemented for our reasoner prototype is closed-world instantiation check.

Some research groups suggest using an epistemic operator called 'K' operator to extend the traditional ALC language [5, 3, 6, 9]. The new extended attributive language is called ALCK. 'K' is read as "is known to be held (by the knowledge base)". We also have researched on the K-operator for our closed-world KB validation purpose. We have implemented two K-operator reasoning services for our reasoner prototype:

K-satisfiability

K-satisfiability checks if there exists at least one individual in KB ABox which is known to instantiate a specific concept description. For example, assume that we want to check the satisfiability of (**K**C). We need to look into the ABox to find a individual 'i', and the fact "i∈C" is known to be held KB. Only when this individual 'i' is exist, **K**C is satisfied. Compared to the open-world satisfiability checking, the reasoning of K-satisfiability needs the ABox information to be involved into the reasoning process, while for open-world satisfiability checking, the ABox information is not necessary.

K-instantiation

K-operator allows partially closing specific concepts and roles in instantiation checking while leaving other part of the KB open. For example, assume that we want to have an instantiation checking on (a∈**K**C⊓B). For the part a∈**K**C, we need to check whether "a∈C" is known to be held by the KB with concept C closed; for the part a∈B, we need to check if "a∈B" is held by the KB with concept B open. Only when both parts return "Yes" answers, the K-instantiation checking

will answer "Yes". Compared to the open-world instantiation checking, K-instantiation checking implements local closed-world reasoning to close some specific part of the KB but keeps the other part of KB in the open-world.

4 Related Work

The syntax and semantics of ALCK has been introduced in DL by F. M. Donini, D. Nardi and their research group. [4] The proposed syntax of ALCK is showed below: (where C and D denote full concepts, A denotes primitive concept(concepts which cannot be reduced further to other concepts combination forms), R denotes a role and p denotes a primitive role(roles which cannot be transformed to other combination forms of other roles).

C, D $\longrightarrow \top \mid \bot \mid$ A $\mid$ C $\mid$ C$\sqcap$D $\mid$ C$\sqcup$D $\mid \exists$R.C $\mid \forall$R.C $\mid$ KC

R $\longrightarrow$ p $\mid$ Kp

The semantics of ALCK can be defined by a interpretation pair (I, W). I = (Δ^I, $\cdot^I$) is a first order interpretation with interpretation domain Δ^I and interpretation function $\cdot^I$. There exist infinite worlds(models) which can be interpreted from the ABox information. W is an abstract set of all these possible worlds or models.

The following equations showed the semantics of how the ALCK syntax elements are interpreted in First Order Logic.

$$\top^{I,W} = \Delta^I$$
$$\bot^{I,W} = \emptyset$$
$$A^{I,W} = A^{I,W} \subseteq \Delta^I$$
$$p^{I,W} = p^{I,W} \subseteq \Delta^I \times \Delta^I$$
$$(\neg \mathrm{C})^{I,W} = \Delta^I \backslash C^{I,W}$$
$$(\mathrm{C} \sqcap \mathrm{D})^{I,W} = C^{I,W} \cap D^{I,W}$$
$$(\mathrm{C} \sqcup \mathrm{D})^{I,W} = C^{I,W} \cup D^{I,W}$$
$$(\exists \mathrm{R.C})^{I,W} = \{\mathrm{a} \in \Delta^I \mid \exists \mathrm{b.(a,b)} \in R^{I,W} \wedge \mathrm{b} \in C^{I,W}\}$$
$$(\forall \mathrm{R.C})^{I} = \{\mathrm{a} \in \Delta^I \mid \forall \mathrm{b.(a,b)} \in R^{I,W} \longrightarrow \mathrm{b} \in C^{I,W}\}$$
$$(\mathbf{K}C)^{I,W} = \bigcap_{J \in W} C^{J,W}$$
$$(\mathbf{K}R)^{I,W} = \bigcap_{J \in W} p^{J,W}$$

As described in [2], primitive concepts are interpreted as subsets of the KB domain Δ^I, and primitive roles are interpreted as individual pairs from Δ^IxΔ^I. The intersect, union, existential and value restriction quantifier are interpreted as set operations on domain Δ^I. The epistemic concept **K**C is interpreted as the set of all individuals which belong to the concept C in all the possible models. "In other words, these objects are definitely known to be members of C. Similarly, an epistemic role **K**p is interpreted as the pairs of individuals that belong to the role p in all possible worlds" [6]

5 Concluding Remarks

In this paper, we propose an approach using closed-world semantics to perform instantiation checking on a KB. For a business KB, the concrete facts which are transformed from database format will be examined by the closed-world instantiation check, to see if they satisfy the validation requirements represented by DL quantifier assertions. The closed-world instantiation check function will lead us to the next stage of the research, to design a global KB consistency check function under closed-world semantics. The research on K-operator is targeting on performing closed-world validation on a partial KB, while keep the rest part of the KB in an open-world setting. This characteristic will be useful for some specific validation scenario of a business KB in future.

References

[1] Fact++ `DL` reasoner. 2008. Avaliable at: `http://owl.man.ac.uk/`.

[2] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider. *Description Logic Hand book*. Cambridge University Press, New York, NY, 2nd edition, 2007.

[3] D. Calvanese, G. D. Giacomo, and D. Lembo. Epistemic first-order queries over description logic knowledge bases. *Faculty of Computer Science, Free University of Bozen-Bolzano*, Bolzano, Italy , 2006.

[4] F. M. Donini, M. Lenzerini, D. Nardi, W. Nutt, and A. Schaerf. An epistemic operator for description logics. *Artificial Intelligence*, 100(1-2):225–274, 1998.

[5] S. Grimm and B. Motik. Closed world reasoning in the semantic web through epistemic operators. *FZI Research Center for Information Technologies at the University of Karlsruhe*, Karlsruhe, Germany, 2005.

[6] S. Grimm, B. Motik, and C. Preist. Matching semantic service descriptions with local closed-world reasoning. *presented at: European Semantic Web Conference*, pages pp.575–589, 2006.

[7] P. Group. Pellet `DL` reasoner. 2008. Avaliable at: `http://pellet.owldl.com/`.

[8] U. Hustadt. Do we need the closed-world assumption in knowledge representation? *Working Notes of the KI'94 Workshop: Reasoning about Structured Objects: Knowledge Representation Meets Databases (KRDB'94)*, pages Document D–94–11, 24–26, 1994.

[9] Y. Katz and B. Parsia. Towards a nonmonotonic extension to owl. *presented at: OWL: Experiences and Directions(OWLED2005)*, 2005.

Fusing Multiple Sensors to Detect Network Traffic Anomalies - A Control Theoretic Model

Wei Lu, Mahsa Kiani, Mahbod Tavallaee and Ali A. Ghorbani

Information Security Center of Excellence
Faculty of Computer Science,
University of New Brunswick, Fredericton, NB E3B 5A3, Canada

Intrusion detection has been extensively studied in the last two decades. However, most existing intrusion detection systems (IDSs) detect a limited number of attack types and report a huge number of false alarms. To improve their performance, a hybrid approach has been proposed recently. A big challenge for constructing such a multi-sensor based IDS is how to make accurate inferences that minimize the number of false alerts and maximize the detection accuracy. We address this issue and propose a control theoretic model which fuses results of two anomaly detection methods, namely non-parametric CUmulative SUM (CUSUM) and EM based clustering using a trust-reputation matrix. The experimental evaluation with the 1999 DARPA intrusion detection evaluation dataset shows that our model can achieve a better performance than the two individual detection sensors as well as the union of the two individual sensors.

I. INTRODUCTION

With the enormous growth of computer networks and the huge increase in the number of applications running on top of it, network security is becoming an important issue. As shown in [1], all computer systems suffer from security vulnerabilities which are both technically difficult and economically costly to be solved by the manufacturers. Therefore, the role of Intrusion Detection Systems (IDSs), as special-purpose devices to detect network anomalies and attacks, is becoming more important.

Generally, IDSs use two fundamental approaches including misuse detection (or the signature based approach) and anomaly detection (or the behaviour based approach). In misuse detection the search for evidence of attacks is based on knowledge accumulated from known attacks. This knowledge is represented by attacks' signatures, which are patterns or sets of rules that can uniquely identify an attack. The pros and cons of misuse detection are completely discussed in [2]. The advantages of signature-based approaches are their good accuracy, low false alarm rate and the fact that they give enough information about the type of detected attacks to the system administrator On the other hand, drawbacks include the difficulty of gathering the required information on the known attacks and keeping it up-to-date with new vulnerabilities.

In anomaly detection, models of normal data are built based on normal traffic, and then the deviation from the normal model will be considered as an attack or anomaly. The main advantage of this approach over misuse detection is that it can detect attempts to exploit new and unforeseen vulnerabilities. It can also help detect "abuse of privileges" attacks that do not actually involve exploiting any security vulnerability. However, this approach has its own shortcomings. The main reported problem is a high false alarm rate, which is caused by two kinds of problems. The first one is the lack of a training dataset that covers all the legitimate areas, and the other one is that abnormal behavior is not always an indicator of intrusions. It can happen as a result of factors such as policy changes or the offering of new services by a site.

In order to overcome these challenges and keep the advantages of misuse detection, some researchers have proposed the idea of hybrid detection. There are currently two ways to achieve this goal, one is sequence based

and the other is parallel based. Sequence based hybrid IDSs apply anomaly detection (or misuse detection) first and misuse detection (or anomaly detection) second [3,4]. Combing the advantages of both misuse and anomaly detection, hybrid IDSs achieve a better performance. However, the sequence based approaches might not provide full coverage for the attack types due to the filtering of malicious (normal) traffic. Also the sequence process will prolong the detection and make real-time detection impossible. In contrast, parallel based hybrid IDSs apply multiple detectors in parallel and make an intrusion decision based on multiple output sources, which provide a wide coverage for intrusions and have the potential to detect previously unknown attacks [5]. One of the biggest challenges for parallel based IDSs is how to make accurate inferences that minimize the number of false alarms and maximize the detection accuracy.

In this paper we propose a control theoretic model in order to address this issue. As illustrated in Figure 1, the general architecture of our detection scheme consists of two major components, namely feature analysis and multi-sensor based IDS. During feature analysis, we define and generate fifteen features to characterize the network traffic behavior, in which we expect the more the number of features, the more accurate the entire network will be characterized. These proposed features are then input to the multi-sensor based IDS, in which many intrusion detectors are fused according to a trust-reputation matrix. The final intrusion decision is given through a fuzzy attacking probability output by the inference model.

The major contributions of this paper include: (1) a formalized model based on dynamic programming for achieving the minimum number of false alarms through self-learning and adaptive capability, (2) a hybrid intrusion detection strategy based on a trust-reputation matrix, and (3) a completed flow based analysis for the 1999 DARPA network traffic dataset using the proposed multi-sensor based IDS.

The rest of the paper is organized as follows. Section II presents the formalized model for our multi-sensor based IDS. Section III introduces the fifteen flow-based features and explains the reasons for selecting them. Section IV provides an overview of the two existing anomaly detection approaches, namely the CUSUM algorithm and the Expectation-Maximization (EM) based clustering algorithm. Section V presents the complete network anomalies analysis for the 1999 DARPA intrusion detection evaluation dataset by using our intrusion inference model. Section 6 makes some concluding remarks and discusses future work.

II. FORMALIZED MODEL FOR MULTI-SENSOR IDS

Figure 2 illustrates the formalized model for the multi-sensor IDS. In particular, the meaning of the notations appearing in Figure 2 is explained as follows:

- ✓ Feature vector is denoted by $F(f_1, f_2 \ldots f_n)$, in which f_i ($i = 1,2,\ldots,n$) refers to features that might be based on flows, packets, host logs, firewall/alert events, traffic behaviour, biometrics, to name a few. In this case, the feature vector denotes the 15-dimensional flow based features.
- ✓ Detection sensors are denoted by S (S_1, S_2, S_3, …,S_m) that include m different detection algorithms for intrusion detection.
- ✓ Notation TRW refers to the Trust-Reputation Weight matrix and it measures the credibility degree of decisions. In particular, TRW_{Sjfi} is the trust-reputation weight for feature f_i in S_j, where $i = 1,2,\ldots,n$ and $j = 1,2,\ldots,m$. The higher the value of TRW_{Sjfi}, the more credible its decision by feature f_i and S_j is. The settings of TRW_{Sjfi} are based on the historical detection records. For each separate feature f_i, we have:

$$\sum_{j=1}^{m} TRW_{S_j f_i} = 1$$

- ✓ $p_{f_i S_j}$ denotes the attacking probability generated by feature f_i and detection sensor S_j. It measures the anomalous degree of current networks by feature f_i and detection sensor S_j, where $i = 1,2,\ldots,n$ and $j = 1,2,\ldots,m$. The higher the value of $p_{f_i S_j}$, the more anomalous the current network. Notation p_{f_i} is the attacking probability correlated by all detection sensors S_j ($j = 1,2,\ldots,m$) with specific features f_i, and we have:

$$p_{f_i} = \sum_{j=1}^{m} p_{f_i S_j} \times TRW_{S_j f_i} \qquad i = 1, 2, ..., n$$

- ✓ $p_{anomalous}$ denotes the final attacking probability generated by MAADS, and we have:

$$p_{anomalous} = \sum_{j=1}^{n} p_{S_j} \times TRW_{S_j}$$

- ✓ Notation *FACount* is the number of false alerts obtained from historical alerting reports. Security officers verify every alert reported by and make after-event decisions on true or false.
- ✓ Based on *FACount*, *penalty factor* and *reward factor* are used to adjust the value of RW_{fiSj} and RW_{Sj} in order to minimize *FACount*.

We conduct a theoretical analysis to prove that the proposed multi-sensor system can always reach the optimum through a dynamic programming technique. For more information about the proof refer to [6].

III. Feature Analysis

The major goal of feature analysis is to select and extract robust network features that have the potential to discriminate anomalous behaviors from normal network activities. Since most current network intrusion detection systems use network flow data (e.g. netflow, sflow, ipfix) as their information sources, we focus on features in terms of flows.

The following five basic metrics are used to measure the entire network's behavior: (1) **FlowCount:** a flow consists of a group of packets going from a specific source to a specific destination over a time period, (2) **PacketCount:** the average number of packets in a flow over a time interval. Most attacks happen with an increased packet count, (3) **ByteCount:** the average number of bytes in a flow over a time interval, (4) **PacketSize:** the average number of bytes per packet over a time interval, and (5) **FlowBehavior:** the ratio between FlowCount and PacketSize. It measures the anomalousness of flow behavior.

Based on the above five metrics, we define a set of features to describe entire traffic behavior on networks. Let F denote the feature space of network flows, a 15-dimensional feature vector $f \in F$ can be represented as $\{f_1, f_2, ..., f_{15}\}$, where the meaning of each feature is explained in Table I.

Empirical observations with the 1999 DARPA network traffic flow logs show that network traffic volumes can be characterized and discriminated through these features. For more information about the results of the empirical observation refer to [6].

IV. Overview of Two Detection Sensors

In this section, we briefly introduce the two network intrusion detection techniques, namely the non-parametric Cumulative SUM (CUSUM) algorithm and the Expectation-Maximization (EM) based clustering algorithm. More information about the CUSUM and EM clustering algorithms can be found in [7] and [8], respectively.

A. Non-parametric CUSUM Algorithm

The CUSUM algorithm is an approach to detect a change of the mean value of a stochastic process. A basic assumption for the non-parametric CUSUM algorithm is that the mean value of the random sequence is negative during normal conditions, and becomes positive when a change occurs. Consequently, a transformation of $\{X_n\}$ into a new sequence $\{Z_n\}$ is necessary, which is given by $Z_n = X_n - \beta$, where β is a constant. The parameter β is set according to network normal conditions and it guarantees that the majority of values of the sequence Z_n are negative during normal conditions and becomes positive when a change occurs. In practice, a recursive non-parametric CUSUM algorithm is used to detect anomalies online. The recursive version is presented in [7,9] and can be defined using a new sequence $\{Y_n\}$:

$$\begin{cases} Y_n = (Y_{n-1} + X_n - \beta)^+ \\ Y_0 = 0 \end{cases} \quad \text{where} \quad x^+ = \begin{cases} x, & x > 0 \\ 0, & otherwise \end{cases}$$

where β is set in a fashion that the values of X_n - β remain slightly negative during normal operations. As a result, increases in the metric are expected to be detected, once the values are bigger than β. A long time period of values larger than β will lead to further increases in the CUSUM function until a possible alarm level is reached. A large value of Y_n is a strong indication of an attack. Based on this, we define an attacking probability p to measure the anomalous degree of the initial sequence X_n:

$$p = \begin{cases} \dfrac{Y_n}{\alpha \times \beta}, & Y_n < \alpha \times \beta \\ 1.0, & otherwise \end{cases}$$

where p is the attacking probability for sequence X_n; α is an adjusting parameter, which is used to amplify the value of β and is set as constant 1, 2,…; Y_n is the CUSUM value of sequence X_n.

B. EM based Clustering Algorithm

The EM algorithm is widely used to estimate the parameters of a Gaussian Mixture Model (GMM). GMM is based on the assumption that the data to be clustered are drawn from one of several Gaussian distributions. It is suggested that Gaussian mixture distributions can approximate any distribution up to an arbitrary accuracy, as long as a sufficient number of components are used. Consequently, the entire data collection is seen as a mixture of several Gaussian distributions, and their corresponding probability density functions can be expressed as a weighted finite sum of Gaussian components with different parameters and mixing proportions. The conditional probability in EM describes the likelihood that data points approximate a specified Gaussian component. The greater the value of conditional probability for a data point belonging to a specified Gaussian component, the more accurate the approximation is. As a result, data points are assigned to the corresponding Gaussian components according to their conditional probabilities. However, in some cases, there exist some data points whose conditional probability of belonging to any component of a GMM is very low or close to zero. These data are naturally seen as the outliers or noisy data. All the outlier data will be deleted or considered as anomalies during anomaly detection, and their attacking probability is set to 1.0. Algorithm I illustrates a detailed EM based clustering algorithm in which C_m stands for the clustering results.

In order to apply the EM based clustering technique for detecting network anomalies, we make two basic assumptions: (1) the input data points are composed of two clusters, namely anomalous cluster and normal cluster; (2) the size of the anomalous cluster is always smaller than the size of the normal cluster. Consequently, we can easily label the anomalous cluster according to the size of each cluster. The attack probability for each data point is equal to the conditional probability of the corresponding data point belonging to the anomalous cluster, which is defined as follows:

$$p = p_{r-1}(C_{anomalous} \mid x_n)$$

where x_n is the data point; $C_{anomalous}$ is the anomalous cluster; $p_{r-1}(C_{anomalous} \mid x_n)$ is the conditional probability of x_n belonging to anomalous cluster $C_{anomalous}$.

V. PERFORMANCE EVALUATION

We evaluate our multi-sensor IDS with the full 1999 DARPA intrusion detection dataset and identify the intrusions based on each specific day. Since most current existing network intrusion detection systems use network flow data (e.g. network, sflow, ipfix, etc.) as their information sources, we convert all the raw TCPDUMP packet data into flow based traffic data by using the public network traffic analysis tools, similar to the 1999 KDDCUP dataset [10] in which the 1998 DAPRA intrusion detection dataset [11] has been converted into a connection based dataset. Although the 1998 and 1999 DARPA dataset was criticized in [12] due to the methodology for simulating an actual network environment, they are a widely used and acceptable benchmark for current intrusion detection research.

During the evaluation, the results are summarized and analyzed in three different categories, namely how many attack instances are detected by each feature and all features' correlation, how many attack types are detected by each feature and all features' correlation and how many attack instances are detected for each attack type. We do not use the traditional Receiver Operating Characteristic (ROC) curve to evaluate our approach and analyze the tradeoff between the false positive rates and detection rates because ROC curves are often misleading and incomplete [13]. Compared to most evaluations with the 1999 DARPA dataset, our evaluation covers all types of attacks and all days' network traffic and thus, we consider our evaluation to be a comprehensive. Next, we will analyze and discuss the intrusion detection results we obtain. More information about the 1999 DAPRA/MIT Lincoln intrusion detection dataset and the method for converting the TCPDUMP packet logs into network flow based logs can be found in [14] and [15], respectively.

A. Creating Trust-Reputation Matrix

The historical reputation matrix is set up according to the detection rate (DR) and the false positive rate (FPR) for each detector over a long time history. The ratio of DR to FPR is used to measure the performance of each detector. We evaluate individually the two detectors with the 15 features and 9 days DARPA testing data on week 4 and week 5. The evaluation results are summarized and analyzed in three different categories described above.

For the detector using the EM based clustering technique, Table II illustrates the average value of DR, FPR and the ratio of DR to FPR for each feature over those 9 days. For the detector using the CUSUM algorithm, Table III illustrates the average value of DR, FPR and the ratio of DR to FPR for each feature. For more information about detection results for features F1 to F15 over 9 days of evaluation using EM based clustering and CUSUM see [6]. Based on the ratio of DR to FPR in Tables II and III, we normalize them and use the normalized values as elements of the historical reputation matrix, which is given as follows:

$$HRW_{2\times 15} = \begin{bmatrix} 0.78 & 0.81 & 0.95 & 0.7 & 0.74 & 0.34 & 0.51 & 0.58 & 0.73 & 0.84 & 0.0 & 0.76 & 0.54 & 0.65 & 0.72 \\ 0.22 & 0.19 & 0.05 & 0.3 & 0.26 & 0.66 & 0.49 & 0.42 & 0.27 & 0.16 & 0.0 & 0.24 & 0.46 & 0.35 & 0.28 \end{bmatrix}$$

The matrix has 2 rows and 15 columns. Row 1 means the historical reputation weight for the detector using the EM based clustering algorithm and row 2 stands for the historical reputation weight for the CUSUM based detector. Columns 1 to 15 stand for the features F1 to F15.

B. Intrusion Detection Results for the Multi-Sensor IDS

The calculation of the attacking probability for the multi-sensor IDS has been discussed in Section 2 in theory. In our evaluation, we substitute real numbers into the generalized model and discuss how to calculate the attacking probability in the system. We have known that there are 15 features and 2 detectors included in the system, and thus *n* is equal to 15; *m* is equal to 2. We define S_1 as the detector using the EM based clustering algorithm and S_2 as the CUSUM based detector. $TRW_{S_jF_i}$ stands for the trust-reputation matrix described in Section 2, where i = 1,2,…,15 and j = 1,2. $p_{F_iS_j}$ is the attacking probability generated by feature F_i and detection agent S_j and p_{F_i} is the attacking probability of the hybrid detection system with specific features F_i. Based on these, we have:

$$p_{F_i} = \sum_{j=1}^{2} p_{F_iS_j} \times HRW_{S_jF_i} \qquad i = 1,2,...,15$$

We evaluated the detection system with one day's DARPA testing data (i.e. W4D1). There are 14 attack types on W4D1 and a total of 8 attack types are detected by our hybrid system. The DR in terms of attack types is 57.14%. The number of attack types detected by using the CUSUM technique is only 5 and the number is 8 with using the EM based clustering only. More detailed detection results regarding the hybrid detection system see [6]. Using S_1 and S_2 to denote the EM based clustering detection sensor and the CUSUM detection sensor, we know the number of correct alerts generated by S_1 is 161; the number of correct alerts generated by S_2 is 73; and the number of correct alerts generated by the hybrid detection system is 105. The intersection set of correct alerts reported by both S_1 and S_2 is 69 and the union set of correct alerts reported by S_1 and S_2 is 166. In order to evaluate and compare the performance of the hybrid system with the two individual detectors, we define two performance metrics, namely degree of agreement and hybrid goodness, which can be calculated as follows:

$$Degree\ of\ Agreement = \frac{Number\ of\ Correct\ Alerts\ in\ Intersection\ Set\ of\ Both\ Detectors}{Number\ of\ Correct\ Alerts\ in\ Union\ Set\ of\ Both\ Detectors} \qquad Hybrid\ Goodness = \frac{Number\ of\ Correct\ Alerts\ Detected\ By\ Hybrid\ System}{Number\ of\ Correct\ Alerts\ in\ Intersection\ Set\ of\ Both\ Detectors}$$

The degree of agreement for the two individual detectors is 0.42. Denoting the intersection of S_1 and S_2 as *S*, the number of correct alerts in the intersection set between *S* and the hybrid system is 49. That is, the hybrid detector reports 49 alerts which are in the same set with the 69 alerts agreed upon by both detectors, and thus, the hybrid goodness of the system in this case is 0.71, which measures the degree of the goodness of the hybrid detection system.

The evaluation results show that the number of correct alerts generated by the hybrid system is 105, which is smaller than the 161 correct alerts generated by the detector using the EM based clustering algorithm. The number of false alerts reported by the hybrid system, however, is 189, which is much smaller than the 799 false alerts by the clustering based detector. That means even though the number of correct alerts reported by the clustering based detector is 1.5 times the number of correct alerts reported by the hybrid detection system, the number of false alerts reported by hybrid is largely reduced, which is only 24% of the total number of false alerts reported by the clustering based detector. From this perspective, we can conclude that our hybrid detection system can achieve an acceptable detection rate but at the same time largely reduce the number of false alerts.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a formalized model for a multi-sensor intrusion detection system. In order to characterize the behaviour of the network flows, we present a 15-dimensional feature vector. The empirical observation results with the 1999 DARPA intrusion detection dataset show that the proposed features have the

potential to distinguish anomalous activities from normal network behaviours. A complete traffic analysis for the 1999 DARPA intrusion detection dataset is conducted using the multi-sensor IDS with two well-known intrusion detection sensors. Based on the achieved evaluation results, we conclude that even though the number of correct alerts reported by the hybrid system is a little bit smaller than the number reported by one of the individual detectors, the hybrid system reduces the number of false alerts largely. Moreover, in this work two new metrics have been proposed to evaluate the performance of the hybrid system, namely *degree of agreement* and *hybrid goodness*.

Future work mainly consists of using more detectors in our system, developing more evaluation metrics to judge the fusion performance and improving the hybrid system through dynamic programming techniques.

REFERENCES

[1] C. E. LANDWEHR, A. R. BULL, J. P. MCDERMOTT, AND W. S. CHOI. A TAXONOMY OF COMPUTER PROGRAM SECURITY FLAWS. ACM COMPUT. SURV., 26(3):211-254, 1994.

[2] H. DEBAR, M. DACIER, AND A. WESPI. TOWARDS A TAXONOMY OF INTRUSION-DETECTION SYSTEMS. COMPUTER NETWORKS: SPECIAL ISSUE ON COMPUTER NETWORK SECURITY, 31(9):805-822, APRIL 1999.

[3] J. ZHANG AND M. ZULKERNINE, A HYBRID NETWORK INTRUSION DETECTION TECHNIQUE USING RANDOM FORESTS. IN PROCEEDINGS OF THE 1ST INTERNATIONAL CONFERENCE ON AVAILABILITY, RELIABILITY AND SECURITY, PP: 262-269, VIENNA UNIVERSITY OF TECHNOLOGY, 2006.

[4] M. QIN, K. HWANG, M. CAI AND Y. CHEN, HYBRID INTRUSION DETECTION WITH WEIGHTED SIGNATURE GENERATION OVER ANOMALOUS INTERNET EPISODES, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, 4 (1), PP: 41-55.

[5] T. SHON AND J. MOON, A HYBRID MACHINE LEARNING APPROACH TO NETWORK ANOMALY DETECTION, INTERNATIONAL JOURNAL ON INFORMATION SCIENCES, VOL. 177, ISSUE 18, PP: 3799-3821, ELSEVIER SCIENCE INC. NEW YORK, 2007.

[6] HTTP://NSL.CS.UNB.CA/WEI/HYBRID.HTM.

[7] H. N. WANG, D. L. ZHANG, AND K. G. HIN. DETECTING SYN FLOODING ATTACKS. IN PROCEEDINGS OF IEEE INFOCOM 2002, JUNE 2002.

[8] W. LU AND I. TRAORE. UNSUPERVISED ANOMALY DETECTION USING AN EVOLUTIONARY EXTENSION OF K-MEANS ALGORITHM. INTERNATIONAL JOURNAL ON INFORMATION AND COMPUTER SECURITY, VOLUME 2, NUMBER 2, PP. 107-139 (33 PAGES), INDERSCIENCE PUBLISHER, MAY 2008.

[9] T. PENG, C. LECKIE, AND K. RAMAMOHANARAO. DETECTING DISTRIBUTED DENIAL OF SERVICE ATTACKS USING SOURCE IP ADDRESS MONITORING. DRAFT, NOVEMBER 2002.

[10] HTTP://KDD.ICS.UCI.EDU/DATABASES/KDDCUP99/KDDCUP99.HTML.KDDCUP

[11] HTTP://WWW.LL.MIT.EDU/IST/IDEVAL/DATA/1998/1998_DATA_INDEX.HTML

[12] M.V. MAHONEY AND P.K. CHAN, AN ANALYSIS OF THE 1999 DARPA/LINCOLN LABORATORY EVALUATION DATA FOR NETWORK ANOMALY DETECTION. IN PROCEEDINGS OF THE 6TH INTERNATIONAL SYMPOSIUM ON RECENT ADVANCES IN INTRUSION DETECTION, PP: 220-237, PITTSBURGH, PA, USA, 2003.

[13] J.E. GAFFNEY, J.W. ULVILA, EVALUATION OF INTRUSION DETECTORS: A DECISION THEORY APPROACH. IN PROCEEDING OF IEEE SYMPOSIUM ON SECURITY AND PRIVACY, PP: 50-61, 2001.

[14] HTTP://WWW.LL.MIT.EDU/IST/IDEVAL/DATA/1999/1999_DATA_INDEX.HTML

[15] W. LU AND A. A. GHORBANI. NETWORK ANOMALY DETECTION BASED ON WAVELET ANALYSIS. EURASIP JOURNAL ON ADVANCES IN SIGNAL PROCESSING, 2008, IN PRESS.

APPENDIX: FIGURES AND TABLES

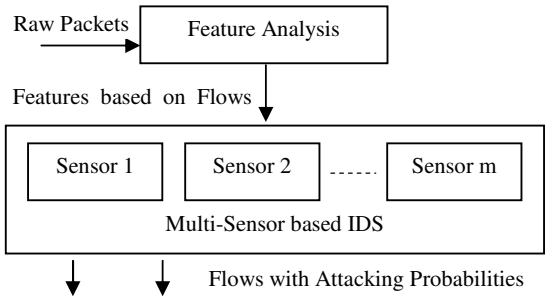


Fig. 1. General architecture of the detection framework

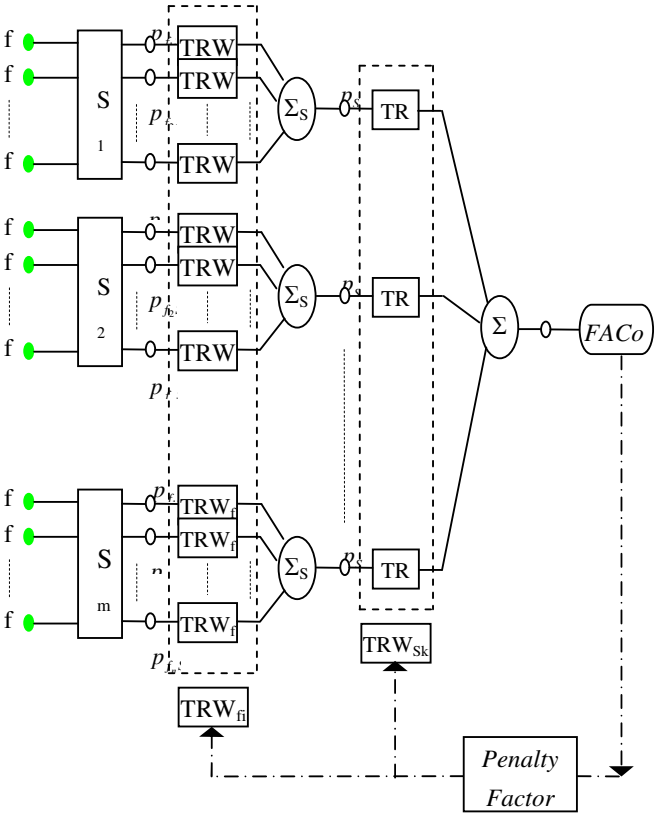


Fig. 2 Formalized model for multi-sensor IDS

ALGORITHM I

EM BASED CLUSTERING ALGORITHM

Function EMCA (data) **returns**
clusters C_m and posterior probability $p_r(i \mid x_n)$
$C_m = \phi$, $1 \le m \le k$, k is the number of clusters
Call EM (data);
For $1 \le m \le k$, $1 \le n \le N$
If ($p_{r-1}(m \mid x_n) = \max(p_{r-1}(m \mid x_n))$)
Then assign x_n to C_m
Return C_m, $m = 1, 2..., k$

TABLE I

LIST OF FEATURES

Features	Description
f_1	Number of TCP Flows per Minute
f_2	Number of UDP Flows per Minute
f_3	Number of ICMP Flows per Minute
f_4	Average Number of TCP Packets per Flow over 1 Minute
f_5	Average Number of UDP Packets per Flow over 1 Minute
f_6	Average Number of ICMP Packets per Flow over 1 Minute
f_7	Average Number of Bytes per TCP Flow over 1 Minute
f_8	Average Number of Bytes per UDP Flow over 1 Minute
f_9	Average Number of Bytes per ICMP Flow over 1 Minute
f_{10}	Average Number of Bytes per TCP Packet over 1 Minute
f_{11}	Average Number of Bytes per UDP Packet over 1 Minute
f_{12}	Average Number of Bytes per ICMP Packet over 1 Minute
f_{13}	Ratio of Number of flows to Bytes per Packet (TCP) over 1 Minute
f_{14}	Ratio of Number of flows to Bytes per Packet (UDP) over 1 Minute
f_{15}	Ratio of Number of flows to Bytes per Packet (ICMP) over 1 Minute

TABLE II

PERFORMANCE OF ALL 15 FEATURES OVER 9 DAYS EVALUATION

Features	**Average DR (%)**	**Average FPR (%)**	**Ratio of Avg. DR to Avg. FPR**
F1	39.83	81.84	0.487
F2	52.22	84.04	0.621
F3	32.25	84.14	0.383
F4	12.0	89.03	0.135
F5	51.8	85.74	0.604
F6	32.25	84.17	0.383
F7	3.2	82.92	0.0386
F8	49.26	84.19	0.585
F9	32.25	84.14	0.383
F10	6.81	86.71	0.0785
F11	0.0	0.0	0.0
F12	32.25	84.17	0.383
F13	8.57	94.59	0.0906
F14	52.41	83.59	0.627
F15	32.25	84.17	0.383

TABLE III

PERFORMANCE OF ALL 15 FEATURES OVER 9 DAYS EVALUATION

Features	**Average DR (%)**	**Average FPR (%)**	**Ratio of Avg. DR to Avg. FPR**
F1	11.04	80.43	0.137
F2	12.96	85.94	0.15
F3	1.6325	87.33	0.02
F4	4.9	84.44	0.058
F5	17.84	82.42	0.217
F6	7.23	95.5	0.757
F7	2.94	79.61	0.037
F8	33.57	78.18	0.429
F9	11.5	81.1	0.142
F10	1.4	94.87	0.015
F11	0.7	95.24	0.0074
F12	10.8	87.26	0.124
F13	6.015	77.18	0.078
F14	27.73	81.85	0.339
F15	12.87	86.12	0.15

An Incremental Self-Improvement Hybrid Intrusion Detection System

Mahbod Tavallaee, Wei Lu, and Ali A. Ghorbani
Faculty of Computer Science, University of New Brunswick
{m.tavallaee,wlu,ghorbani}@unb.ca

Abstract

Combining misuse and anomaly detection methods into a hybrid system has been recently proposed in order to improve intrusion detection capability. However, there exist two important issues that make this task cumbersome. First, all anomaly-based methods need a completely labeled and up-to-date training set which is very costly and time-consuming to create if not impossible. Second, getting different detection technologies to interoperate effectively and efficiently becomes a big challenge for building an operational hybrid intrusion detection system (IDS).

In this paper , we propose a new hybrid network intrusion detection framework, combining the well known Snort as a signature based detector and a decision tree algorithm (C4.5) as an anomaly detector. Based on the idea of incremental learning, we provide our hybrid system with an automatically labeled training set. This training set will be improved and updated gradually; therefore, probable changes in the traffic behavior will not affect our system. In addition, taking advantage of a fast classifier (C4.5) and simple flow-based features, our hybrid detector can perform real-time with an acceptable delay similar to Snort. Experimental evaluations on real traffic from a large-scale WiFi ISP network show that our approach successfully detects a large portion of the attacks missed by Snort while also reducing the false alarm rate.

1. Introduction

Intrusion detection has been extensively studied since the seminal work by Anderson [1]. Traditionally, intrusion detection techniques are classified into two categories: misuse (signature-based) detection and anomaly detection. Misuse detection is based on the assumption that most attacks leave a set of signatures in the stream of network packets or in audit trails, and thus attacks are detectable if these signatures can be identified by analyzing the audit trails or network traffic behaviors. However, misuse detection is strictly limited to the known attacks and detecting new attacks is one of the biggest challenges faced by misuse detection.

To address the weakness of misuse detection, the concept of anomaly detection was formalized in the seminal report of Denning [2]. In this approach models of normal data are build based on the normal traffic, and then the deviation from the normal model will be considered as an attack or anomaly. The main advantage of this approach over misuse detection is that it can detect attempts to exploit new and unforeseen vulnerabilities. It also can help detect "abuse of privileges" types of attacks that do not actually involve exploiting any security vulnerability. However, this approach has its own shortcomings. The main reported problem is high false alarm rate which is caused by two kinds of problems. The first one is the lack of a training data set that covers all the legitimate areas, and the other one is that abnormal behavior is not always an indicator of intrusions. It can happen as a result of factors such as policy changes or offering of new services by a site.

In order to overcome these challenges, and keep the advantages of misuse detection, some researchers have proposed the idea of hybrid detection. This way, the system will achieve the advantage of misuse detection to have a high detection rate on known attacks as well as the ability of anomaly detectors in detecting unknown attacks. According to this fusion approach,

current hybrid IDSs can be divided into two categories: 1) **sequence-based** in which either anomaly detection or misuse detection is applied first, and the other one is applied next; 2) **parallel-based** in which multiple detectors are applied in parallel, and the final decision is made based on multiple output sources.

Although with respect to the characteristics of signature-based and anomaly-based methods, the fusion of these two approaches should theoretically provide a high-performance IDS, there are still two important issues that make this task cumbersome. First, all anomaly-based methods need a completely labeled and up-to-date training set which is very costly and time-consuming to create if not impossible. Second, getting different detection technologies to interoperate effectively and efficiently becomes a big challenge for building an operational hybrid intrusion detection system.

To overcome the aforementioned problems, we have combined a signature-based detector (Snort [5]) and an anomaly-base detector (C4.5 [3]) in parallel with the idea of incremental learning. Toward this aim we have defined learning time intervals, e.g. 1 day, at the end of which the anomaly-based detector will be trained by the latest training set. This training set is the flows labeled by the hybrid detector in the previous interval. In the first interval which we do not have any training set, we only rely on the labels from Snort. These labels will be used as a training set for the anomaly-based detector in the next time interval. During the second time interval raw packets are given to both Snort to do the labeling and Flow Aggregator to provide the required features for C4.5 Classifier. The flow-based features will then go through the decision tree based classifier to be labeled. Finally, we use a fusing algorithm to combine the results from Snort and the Classifier. This result will be both reported to the admin and used as a training set in the next time interval.

Although at first we only rely on the Snort labels which are not very accurate to be used as a training set, fusion of Snort and the C4.5 classifier will provide a more reliable training set as we go forward. This way we hope to gain a pretty reliable system after running the system for a while. The other advantage of our method is that the training set will be changed as the traffic behavior changes and keeps itself up-to-date. Applying a decision tree based classifier which has a very low classification time on the one hand, and applying flow-based features computed on-line on the other hand, makes our system completely real-time.

The major contributions of this paper include: 1) a novel method to provide the system with an automatically labeled training set; 2) defining some time intervals to change the training set and replacing it with the latest one, and keeping the training set up-to-date; 3) providing a general framework to combine signature-based and anomaly-based detectors together in order to achieve the advantages of anomaly detectors while keeping all the benefits of misuse detection; 4) taking advantage of a fast classifier and simple flow-based features to keep the hybrid detector real-time with an acceptable delay similar to Snort.

The rest of the paper is organized as follows. Our proposed detection scheme will be explained in Section 2. Section 3 presents the experimental evaluation of our approach and discusses the obtained results. Finally, in Section 4, we draw conclusions.

2. The Proposed Detection Scheme

As mentioned in Section 1 with respect to the characteristics of signature-based and anomaly-based methods, the fusion of these two approaches effectively should theoretically provide a

high-performance IDS. However, there are two important issues that make this task cumbersome. First, all anomaly-based methods need a completely labeled and up-to-date training set which is very costly and time-consuming to create if not impossible. Second, getting different detection technologies to interoperate effectively and efficiently becomes a big challenge for building an operational hybrid intrusion detection system (IDS)

In the rest of this section we briefly explain the anomaly-based and signature-based detectors we have applied, and then will provide our solutions to solve the aforementioned problems.

A. Anomaly-based Detector

As the first step to have an effective anomaly detector, we should extract robust network features that have the potential to discriminate anomalous behaviors from normal network activities. Since most current network intrusion detection systems use network flow data (e.g. netflow, sflow, ipfix) as their information sources, we focus on features generated based on these flows. The name and description of the applied features are listed in Table 1.

In order to create the flows and extract the features we used a commercial network security management tool called QRadar [4]. In addition to provide statistical features this product has the functionality of detecting the type of applications in each flow. Running some experiments we found this feature quite helpful to increase the performance of the system. In addition, this feature is calculated on-line and does not impose any delays to the system.

Having extracted the features, the next step is to find a very efficient classifier. Evaluating famous classifiers based on detection rate, false alarm rate, classification time, and learning time, we ended up with the C4.5 decision tree algorithm [3].

Table 1. Applied flow-based features

SrcIP	source IP address
DstIP	destination IP address
SrcPort	source port number
DstPort	destination port number
SrcBytes	number of bytes in the flow sent from the source to the destination
DstBytes	number of bytes in the flow sent from the destination to the source
SrcPackets	number of packets in the flow sent from the source to the destination
DstPackets	number of packets in the flow sent from the destination to the source
SrcBytes/DstBytes	the ratio of “SrcBytes” to “DstBytes”
SrcPackets/DstPackets	the ratio of “SrcPackets” to “DstPackets”
SrcBytes/SrcPackets	the ratio of “SrcBytes” to “SrcPackets”
DstBytes/DstPackets	the ratio of “DstBytes” to “DstPackets”
Protocol Name	Value of “protocol” field in the IP packet
Application Name	Name of the application detected by an “application discovery” module

B. Signature-based Detector

As our signature-based detector we chose Snort [5] because of its popularity and availability to researchers. However, our proposed hybrid detection scheme is completely independent from Snort, and any other signature-based detector can be used instead.

As mentioned earlier, our anomaly-based detector works on flows. However, Snort is designed to work on packets. To make our detectors consistent, we matched snort alerts with the existing flows based on the source IP, source port, destination IP, destination port, and time stamp. Since

the flows and snort alerts are generated by different devices, we were not very strict with the time stamps and considered a deviation of up to 5 seconds acceptable.

C. The proposed hybrid detector

The most important issues that current anomaly detectors deal with are firstly to prepare a labeled data set, and secondly to keep that data set up-to-date. To solve these problems we have proposed to apply the idea of incremental learning. To meet this goal we have defined learning time intervals, e.g. 1 day, at the end of which the anomaly-based detector will be trained by the latest training set. This training set is the flows labeled by the hybrid detector in the previous interval. Figure 1 illustrates the structure of our hybrid detector. In the first interval which we do not have any training set we only rely on the labels from Snort. These labels will be used as a training set for the anomaly-based detector in the next time interval. During the second time interval, raw packets are given to both Snort to do the labeling and Flow Aggregator to provide the required features for C4.5 Classifier. The flow-based features will then go through the decision tree based classifier to be labeled. Finally, we use a fusing algorithm to combine the results from Snort and the Classifier. This result will be both reported to the administrator and used as a training set in the next time interval.

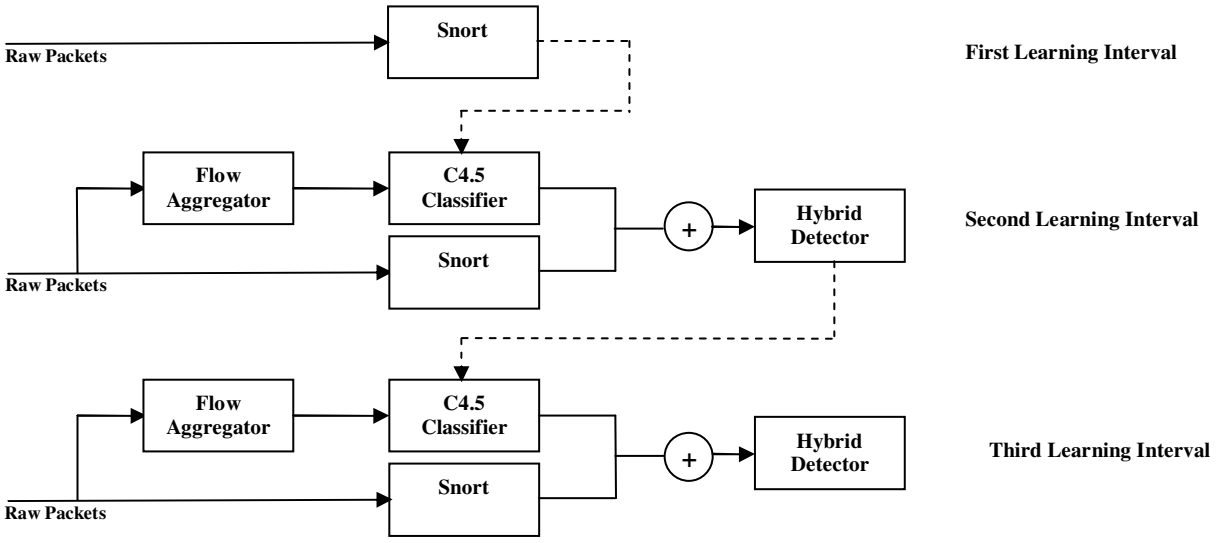


Figure 1. General structure of the hybrid detector

D. Fusing algorithm

Let $L_S^i(t)$ and $L_C^i(t)$ be the labels assigned to the ith flow of time interval t by Snort and the classifier respectively; N denote normal traffic, and a_j^t denote an attack of type j in time interval t. Using Algorithm 1 the hybrid detector will specify the final label of each flow.

As illustrated in Algorithm 1, when the flow is labeled as normal by both Snort and classifier, it will be labeled as normal by the hybrid detector. Similarly, if the flow is labeled as an attack by both detectors, it will be labeled as an attack by the hybrid system. However, since signature based method are more accurate in providing the detail of the attacks compared to anomaly

detectors, we choose the attack type detected by Snort. In addition, if the flow is labeled as normal by Snort but as an attack by the classifier, we will label it as an attack since the classifier's knowledge is based on previous detection of Snort, and it is very probable that Snort misses some signatures in the flows. The most complicated situation happens when Snort labels a flow as an attack, while the classifier labels it as normal. In this situation, we look for that specific attack type in the list of attacks existing in the previous training set. If we can find that attack type in the list, it means that the classifier has already learned it; therefore the hybrid detector relies on the classifier and labels that flow as normal. Otherwise, if we cannot find that attack type in the attack list, it shows the classifier does not know anything about it. So, the hybrid system trusts Snort and labels that flow as an attack. This attack type will be learned by the classifier in the next time interval and makes later detection of this kind of attack more accurate in the future.

Alogorithm 1. Fusing Algorithm

Function HybridDetector
  Inputs:
    Collection of flows labeled by Snort $L_S^i(t)$, $i = 1, 2, \dots, m$
    Collection of flows labeled by Classifier $L_C^i(t)$, $i = 1, 2, \dots, m$
    List of existing attack in the previous time interval
      $A(t-1) = \{a_1^{t-1}, a_2^{t-1}, \dots, a_K^{t-1}\}$
  Initialization:
    $i \leftarrow 0$
  Repeat: $i \leftarrow i + 1$
    If $L_S^i(t) = N$ *and* $L_C^i(t) = N$ **Then** $L^i(t) = N$
    Else If $L_S^i(t) = N$ *and* $L_C^i(t) = a_j^t$ **Then** $L^i(t) = a_j^t$
    Else If $L_S^i(t) = a_j^t$ *and* $L_C^i = N$ **Then**
      If $a_j^t \in A(t-1)$ **Then** $L^i(t) = N$ **Else** $L^i(t) = a_j^t$
    Else If $L_S^i(t) = a_j^t$ *and* $L_C^i(t) = a_k^t$ **Then** $L^i(t) = a_j^t$
  Until: $i = m$
Return $L(t)$

3. Experiments

A. Applied data sets

To analyze the performance of our method, we used real traffic from a large-scale WiFi ISP network, Fred-eZone [6], over three consecutive days. Fred-eZone is a free WiFi service which is provided by City of Fredericton, New Brunswick, Canada and covers downtown business districts, City parks, local arenas, business hotels, etc. Table 2 summarizes the workload of the Fred-eZone network.

Table 2. Workload of Fred-eZone WiFi network over 1 day

SrcIP	DstIP	Flows	Packets	Bytes
1055K	1228K	30783K	994M	500G

In order to find the real labels of flows (anomalous or normal) in the second data set, we relied on a commercial product, QRadar [4]. This product includes a complete set of information from

the packets, flows, network characteristics, etc. as an input to their rule engine, and then based on some expert knowledge decides if a flow is anomalous or normal. Although the labels provided by this product are not fully accurate, since it brings a lot of information into account for its final decision, it is much more accurate and reliable than Snort.

B. Experimental Result

To perform our experiment on real traffic from the Fred-eZone data set we chose three consecutive days of traffic. We then divided the traffic into three one-day time intervals. Figures 2 and 3 compare the detection rate and false positive rate of the Snort with our proposed hybrid system, respectively.

As it is illustrated in Figures 2 and 3, our proposed hybrid detector has improved the performance of Snort in terms of both detection rate and false alarm rate.

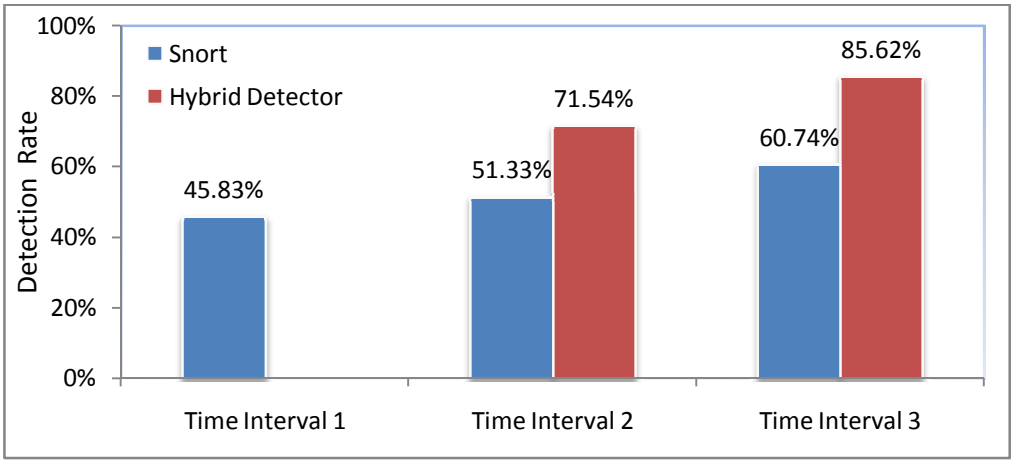


Figure 2. Comparison of Snort and the Hybrid Detector based on the detection rate

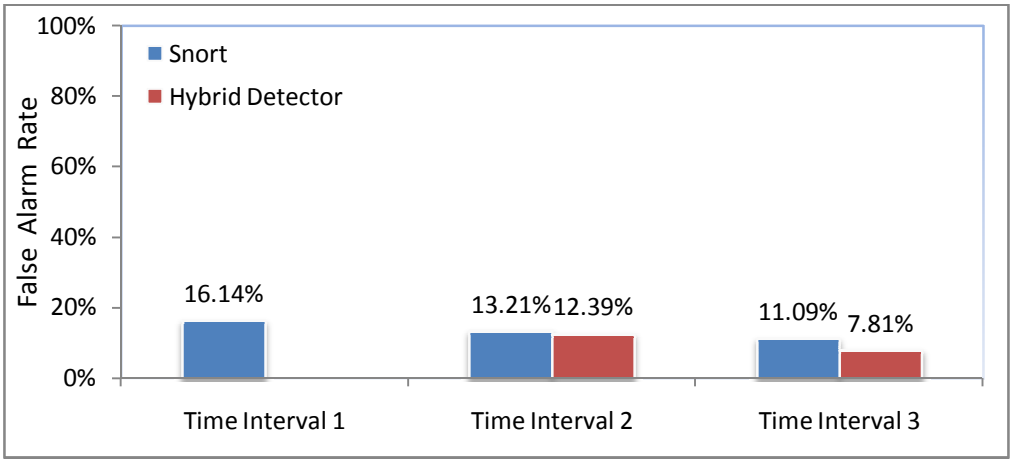


Figure 3. Comparison of Snort and the Hybrid Detector based on the false alarm rate

4. Conclusion

In this paper, we proposed a new hybrid network intrusion detection framework, combining the well known Snort as a signature based detector and a decision tree algorithm (C4.5) as an

anomaly detector. Based on the idea of incremental learning we provided our hybrid system with an automatically labeled training set. This training set will be improved and updated gradually; therefore, probable changes in the traffic behavior will not affect our system. In addition, taking advantage of a fast classifier (C4.5) and simple features, our hybrid detector can perform real-time with an acceptable delay similar to Snort. Experimental evaluations on real traffic from a large-scale WiFi ISP network showed that our approach successfully detected a large portion of the attacks missed by Snort while reducing the false alarm rate. Although in this work we have used Snort as a misuse detector and C4.5 classifier as an anomaly detector, our approach is not restricted to these detectors and can be used as a general framework to combine any misuse and anomaly detection systems.

References

[1] J. P. Anderson. Computer Security Threat Monitoring and Surveillance. *Technical Report, James P. Anderson Co.*, Fort Washington, Pennsylvania, 1999.

[2] D. E. Denning. An Intrusion Detection Model. *IEEE Transactions on Software Engineering*, 2: 222-232, 1987.

[3] J. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.

[4] Q1 Labs network security management company, Available on: http://www.q1labs.com, September, 2008.

[5] Snort: The open source network intrusion detection system. Available on: http://www.snort.org/, August, 2008.

[6] Fred-eZone WiFi Service, Available on: http://www.fred-ezone.com, September, 2008.

Expressing Vague Knowledge in the Fuzzy Description Logic $f\mathcal{ALCHIN}$

Jidi Zhao, Harold Boley†, and Weichang Du
Faculty of Computer Science,
University of New Brunswick, Fredericton, Canada
{Judy.Zhao, wdu} AT unb.ca
† Institute for Information Technology, National Research Council of Canada
Fredericton, NB, E3B 9W4 Canada
Harold.Boley AT nrc.gc.ca

March 26, 2009

Abstract

Uncertainty is an intrinsic feature of our knowledge, which is also reflected in the World Wide World and the Semantic Web. Motivated by Web applications, this paper introduces an expressive fuzzy description logic that extends classical description logics to many-valued logics. The syntax to represent imprecise or vague knowledge and the semantics to interpret complex concept descriptions and subsummptions are addressed in detail. This proposed fuzzy description logic, $f\mathcal{ALCHIN}$, extends the expressiveness of the well know description logic $\mathcal{ALC}$ by fuzzy concepts, fuzzy inverse roles, and fuzzy role inclusion axioms, as well as fuzzy at-most/at least number restrictions.

1 Introduction

The Semantic Web initiative aims at creating an extension of the current World Wide Web by developing logic-based standards and technologies that enable machines to understand the information on the web, so that they can support richer knowledge discovery and automate the performance of various tasks for human beings [3].

A key research direction for the Semantic Web is to handle uncertainty, as evidenced by Fuzzy RuleML [4] and W3C's Uncertainty Reasoning for the World Wide Web Incubator Group [8]. Typical Description Logics (DL) are limited to dealing with crisp, well defined concepts. They cannot express vague or uncertain knowledge. However, uncertainty is an intrinsic feature of real-world knowledge. Many concepts needed in knowledge modeling lack well-defined boundaries or, precisely defined criteria of relationships with other concepts. For example, the concepts of young man, tall, and cold.

To overcome this deficiency, this paper proposed an extension to Description Logics based on Fuzzy Logic. The rest of this paper is organized as follows. Section 2 briefly introduces the syntax and semantics of expressive Description Logics. Section 3 reviews Fuzzy Logic and Fuzzy Set Theory. Section 4 presents the syntax and semantics of an expressive fuzzy description logic, as well as the components of a knowledge base using such this knowledge representation formalism. Section 5 reviews some related work in uncertainty management in Description Logic. Finally, in Section 6 we summarize our main results and give an outlook on future research.

2 Preliminaries

We briefly introduce Description Logics in the current section. Their syntax and semantics in terms of classical First Order Logic are also presented. As a notational convention, we will use a, b, x for individuals, A for atomic concepts, C and D for concept descriptions, R and P for atomic roles.

Description Logics (DL) [2][1] are a family of logic-based knowledge representation formalisms designed to represent and reason about the knowledge of a concrete domain. Elementary descriptions of DL are atomic concepts and atomic roles. Complex concept descriptions can be built from the elementary constructors and construction rules. Different description languages of DL are distinguished by the constructors they provide. For example, $\mathcal{ALCHIN}$ DL extends the well known $\mathcal{ALC}$ DL with inverse roles, role inclusion axioms, and number restrictions. Concept constructors in $\mathcal{ALCHIN}$ are formed according to the syntaxes in Table 1.

Table 1: Syntax and Semantics of $\mathcal{ALCHIN}$ constructors

DL Constructor	DL Syntax	Semantics
top concept	$\top$	$\triangle^I$
bottom concept	$\bot$	$\emptyset$
atomic concept	A	$A^I \subseteq \triangle^I$
concept name	C	$C^I \subseteq \triangle^I$
atomic negation	$\neg A$	$\triangle^I \setminus A^I$
concept negation	$\neg C$	$\triangle^I \setminus C^I$
concept conjunction	$C \sqcap D$	$C^I \cap D^I$
concept disjunction	$C \sqcup D$	$C^I \cup D^I$
exists restriction	$\exists R.C$	$\{x \in \triangle^I \mid \exists y. < x, y > \in R^I \wedge y \in C^I\}$
value restriction	$\forall R.C$	$\{x \in \triangle^I \mid \forall\ y. < x, y > \in R^I \rightarrow y \in C^I\}$
inverse role	R^-	$(R^-)^I(y, x) = R^I(x, y)$
at-most restriction	$\leq nR$	$\{x \in \triangle^I \mid \sharp\{y \in \triangle^I \mid R^I(x, y)\} \leq n\}$
at-least restriction	$\geq nR$	$\{x \in \triangle^I \mid \sharp\{y \in \triangle^I \mid R^I(x, y)\} \geq n\}$

Description Logics have a model theoretic semantics, which is defined by interpreting concepts as sets of individuals and roles as sets of pairs of individuals. An interpretation

I is a pair $I = (\Delta^I, \cdot^I)$ consisting of a domain Δ^I which is a non empty set and of an interpretation function $\cdot^I$ which maps each individual x into an element of Δ^I ($x \in \Delta^I$), each concept C into a subset of Δ^I ($C^I \subseteq \Delta^I$) and each atomic role R into a subset of $\Delta^I \times \Delta^I$ ($R \subseteq \Delta^I \times \Delta^I$). The interpretations of complex concept descriptions are shown in Table 1.

A knowledge base (KB) based on DL $KB =< T, A >$ consists of two parts: the terminological box (TBox T) and the assertion box (ABox A). There are two kinds of assertions in the ABox of a DL KB: concept individual and role individual. A concept instance assertion has the form $C(a)$ while a role instance assertion is $R(a, b)$. The semantics of assertions is interpreted as the assertion $C(a)$ (resp. $R(a, b)$) is satisfied by I iff $a^I \in C^I$ (resp. $(a^I, b^I) \in R^I$).

A DL KB has several kinds of axioms. A concept inclusion axiom is an expression of subsumption with the form $C \sqsubseteq D$. The semantics of a concept inclusion axiom is interpreted as the axiom is satisfied by I iff $\{x \in \Delta^I | \forall x, x \in C^I \rightarrow x \in D^I\}$. A concept equivalence axiom is an expression of the form $C \equiv D$. Its semantics is that the axiom is satisfied by I iff $\{x \in \Delta^I | \forall x, x \in C^I \rightarrow x \in D^I, x \in D^I \rightarrow x \in C^I\}$. An inverse role axiom is of the form $R^- \equiv R$ with the semantics interpreted as the axiom is satisfied by I iff $\{x, y \in \Delta^I | (R^-)^I(y, x) = R^I(x, y)\}$. An role inclusion axiom has the form $R \sqsubseteq P$ with its semantic states that the axiom is satisfied by I iff $\{x, y \in \Delta^I | R^I(x, y) \rightarrow P^I(x, y)\}$. Similarly, we can define the syntax of a role equivalence axiom as $R \equiv P$ and its semantics.

3 Fuzzy Set Theory and Fuzzy Logic

Fuzzy set theory was first introduced by Zadeh [17] as an extension of the classical notion of set to capture the inherent vagueness (the lack of crisp boundaries of sets). Fuzzy logic is a form of multi-valued logic derived from fuzzy set theory to deal with reasoning that is approximate rather than precise. Just as in fuzzy set theory the set membership values can range between 0 and 1, in fuzzy logic the degree of truth of a statement can range between 0 and 1 and is not constrained to the two truth values true, false as in classic predicate logic [10]. Formally, a fuzzy set X with respect to a set of elements Ω (also called a universe) is characterized by a membership function $\mu(x)$ which assigns a value in the real unit interval [0,1] to each element x in X ($x \in X$). $\mu(x)$ gives us an estimation that an element x belongs to a set X to a certain degree. Such degrees could be computed based on some specific membership functions. Figure 1 summarizes the most frequently used crisp, trapezoidal, triangular, left-shoulder, and right-shoulder membership functions. Here we define these functions as *crisp(a,b)*, *leftshoulder(a,b)*, *rightshoulder(a,b)*, *triangular(a,b,c)*, and *trapezoidal(a,b,c,d)* respectively. The domain of these membership functions are defined as $[k_1, k_2]$.

For example, a fuzzy set *Young* is defined by a left_shoulder membership function *leftshoulder(30,50)* as shown in Figure 2. Now we know, John is 34 years old. Therefore, we have *Young(John)=0.8* which means the statement "John is a young man" has a truth value of 0.8. But more often, we want to make vaguer statements, saying that "John is a young man" has a truth value of greater than or equal to 0.8. Such a statement can be

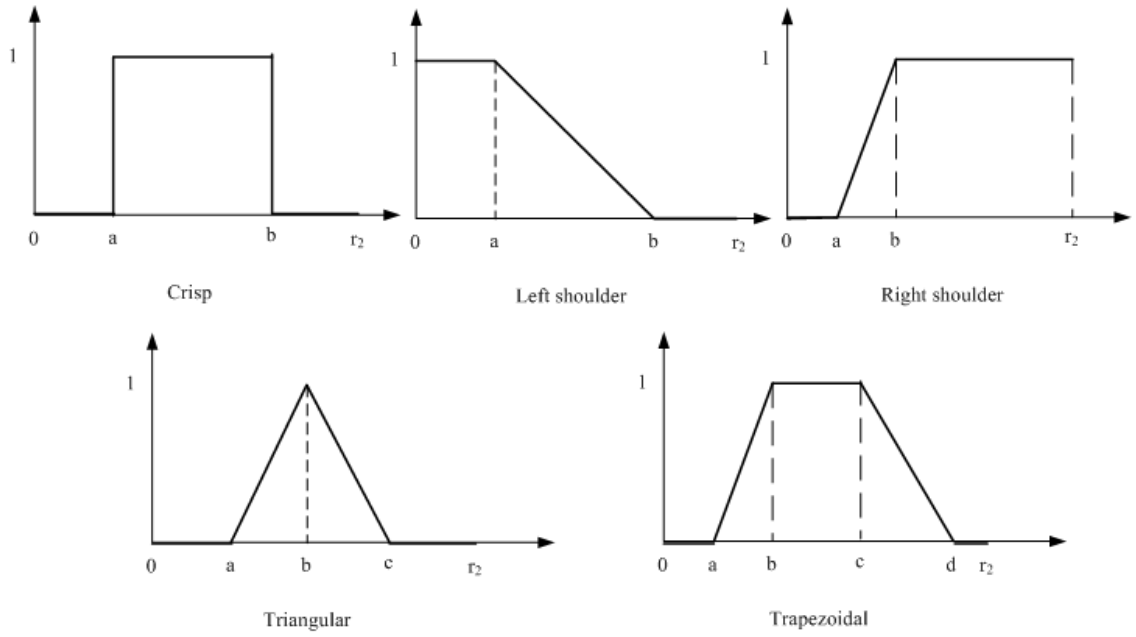


Figure 1: An example of propagating logical consequences in the interface-based modularization formalism

written as *Young(John)≥ 0.8*. Another kind of mainly used statement is less than or equal to. In order to describe all the above statements in a unified form, we propose a syntax as $[l, u](0 \leq l \leq u \leq 1)$. Therefore, $Young(John) \geq 0.8$ can be written as $Young(John)$ $[0.8, 1]$ and $Young(John) \leq 0.8$ as $Young(John)$ $[0, 0.8]$.

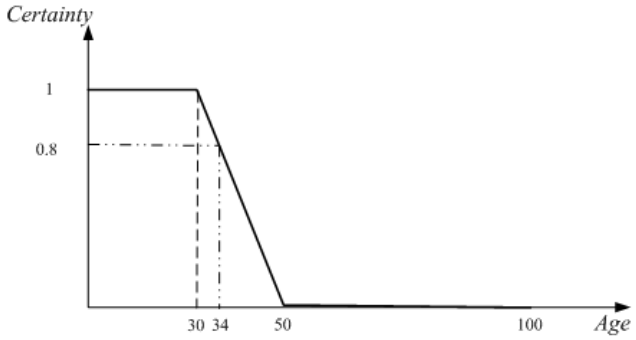


Figure 2: An example of propagating logical consequences in the interface-based modularization formalism

A fuzzy relation R is over two fuzzy sets X_1 and X_2 is defined by a function $R : \Omega \times \Omega \rightarrow [0, 1]$. For example, the statement "Young people drive fast" has a truth value of greater than or equal to 0.6 can be defined as a fuzzy relation R over two fuzzy sets $Young$ and $Fast : R(John, 150)$ $[0.6, 1]$.

Fuzzy logic extends the Boolean operations such as complement, union, and intersection, defined on crisp sets and relations in the context of fuzzy sets and fuzzy relations. These operations are interpreted as mathematical functions over the unit interval [0,1]. The math-

ematical functions for fuzzy intersection are usually called t-norms, those for fuzzy union are called s-norms, and the fuzzy set complement is called negation. Different types of such operations in Fuzzy Logic including Zadeh Logic, Lukasiewicz Logic, Product Logic, and Gödel Logic, are summarized in Table 2. All these operations satisfy certain mathematical properties.

Table 2: Fuzzy Operations

	Zadeh	Lukasiewicz Logic	Product Logic	Gödel Logic
t-norms ($t(x,y)$)	$\min(x,y)$	$\max(x+y-1,0)$	$x \cdot y$	$\min(x,y)$
s-norms ($s(x,y)$)	$\max(x,y)$	$\min(x+y,1)$	$x+y-x \cdot y$	$\max(x,y)$
negation ($\neg x$)	$1-x$	$1-x$	if x=0 then 1 else 0	if x=0 then 1 else 0

4 Fuzzy Description Logic

4.1 Syntax of $f\mathcal{ALCHIN}$

Concept descriptions in $f\mathcal{ALCHIN}$ are formed based on the following syntax:

$C \rightarrow \top|\bot|A|C|\neg A|\neg C|C \sqcap D|C \sqcup D|\exists R.C|\forall R.C|\geq nR|\leq nR$

We can see that the syntax of this fuzzy description logic is identical to that of the standard description logics. But here in $f\mathcal{ALCHIN}$, the concepts and roles are defined as fuzzy concepts (i.e. fuzzy sets) and fuzzy roles (i.e. fuzzy relations).

4.2 Semantics of $f\mathcal{ALCHIN}$

Similar to classical DL, the semantics of the proposed $f\mathcal{ALCHIN}$ is based on the notion of interpretation. Classical interpretations is extended to the concept of fuzzy interpretations by using membership functions that range over the interval [0,1]. An fuzzy interpretation I is a still pair $I = (\Delta^I, \cdot^I)$ consisting of a domain Δ^I which is a non empty set and of a fuzzy interpretation function $\cdot^I$ which maps each individual x into an element of Δ^I ($x \in \Delta^I$), each concept C into a membership function of $C^I : \Delta^I \rightarrow [0,1]$, and each atomic role R into a membership function of $R^I : \Delta^I \times \Delta^I \rightarrow [0,1]$.

Next we define the semantics of $f\mathcal{ALCHIN}$ constructors, including the top concept, the bottom concept, concept negation, concept conjunction, concept disjunction, role exists restriction, role value restriction, and number restrictions. We explain how to apply the fuzzy logic operations in Table 2 to the proposed $f\mathcal{ALCHIN}$ with some examples.

The semantics of the top concept $\top$ is the greatest element in the domain Δ^I, that is, $\top^I = 1$ $(\forall x, x \in \Delta^I)$. Please note that, in classical DL, the top concept $\top \equiv A \sqcup \neg A$, while

in $f\mathcal{ALCHIN}$, $\top \neq A \sqcup \neg A$. As shown in Table 2, after applying the s-norms on $A \sqcup \neg A$, the result is no longer 1, which is contradictory to intuition. Therefore, we explicitly define the top concept, stating that the truth degree of x in $\top$ is 1. Similarly, the bottom concept $\bot$ is the least element in the domain, defined as $\bot^I = 0$ $(\forall x, x \in \Delta^I)$.

The concept negation (also known as concept complement) $\neg C$ is interpreted with a mathematical function which satisfies

1. $\neg\top^I(x) = 0, \neg\bot^I(x) = 1$.
2. self-inverse, i.e., $(\neg\neg C)^I(x) = C^I(x)$.

For example, if we have the statement "John is a young man" has a truth value of greater than or equal to 0.8 ($Young(John)$ $[0.8, 1]$), and assume we choose the negation operator in Zadeh logic or Lukasiewicz logic, then the statement "John is not a young man" is written as $\neg Young(John) = \neg[0.8, 1] = [0, 0.2]$.

The interpretation of concept conjunction (also called concept intersection) is defined by t-norms as

$(C \sqcap D)^I(x) = t(C^I(x), D^I(x))$ $(\forall x, x \in \Delta^I)$

For example, if we have $Young(John)$ $[0.8, 1]$ and $Tall(John)$ $[0.7, 1]$, and assume the minimum function is chosen as the t-norm, then the certainty that John is both young and tall is $(Young \sqcap Tall)(John) = \min([0.8, 1], [0.7, 1]) = [0.7, 1]$.

The interpretation of concept disjunction/union is defined by the s-norms as

$(C \sqcup D)^I(x) = s(C^I(x), D^I(x))$ $(\forall x, x \in \Delta^I)$

For example, if we have $Young(John)$ $[0.8, 1]$ and $Tall(John)$ $[0.7, 1]$, and the s-norm is maximum, then the certainty that John is either young or tall is $(Young \sqcup Tall)(John) = \max([0.8, 1], [0.7, 1]) = [0.8, 1]$.

The semantics of role exists restriction $\exists R.C$ is the result of viewing $\exists R.C$ as the open first order formula $\exists y.F_R(x, y) \wedge F_C(y)$ and the existential quantifier $\exists$ is viewed as a disjunction over the elements of the domain. Therefore, we define

$(\exists R.C)^I(x) = \sup_{y \in \Delta^I}\{t(R^I(x, y), C^I(y))\}$

Suppose we have $hasVitalDisease(John, Cancer)$ $[0.2, 1]$, $VitalDisease(Cancer)$ $[0.5, 1]$, $hasVitalDisease(John, Cold)$ $[0.6, 1]$, and $VitalDisease(Cold)$ $[0.1, 1]$. Further we assume the minimum function is chosen as the t-norm, then

$$\begin{aligned}(\exists R.C)^I(x) &= \sup\{\min(hasVitalDisease(John, Cancer), VitalDisease(Cancer)),\\ &\qquad \min(hasVitalDisease(John, Cold), VitalDisease(Cold))\}\\ &= \sup\{\min([0.2, 1], [0.5, 1]), min([0.6, 1], [0.1, 1])\}\\ &= \sup [0.2, 1], [0.1, 1] = [0.2, 1]\end{aligned}$$

That is, the truth degree for the complex concept assertion $(\exists\, hasVitalDisease.VitalDisease)$ $(John)$ is greater than or equal to 0.2.

A role value restriction $\forall R.C$ is viewed as an implication of the form $\forall y \in \Delta^I, R^I(x, y) \rightarrow C^I(x)$. As proposed by Hajek [5], we interpret $\forall$ as inf. Furthermore, in classical logic, $a \rightarrow b$ is a shorthand for $\neg a \vee b$, we can thus interpret $\rightarrow$ as the Kleene-Dienes implication and finally get its semantics as $(\forall R.C)^I(x) = \inf_{y \in \Delta^I}\{s(\neg R^I(x, y), C^I(y))\}$.

A fuzzy at-least restriction is of the form $\geq nR$ whose semantic

$(\geq nR)^I(x) = \sup_{y_1,\ldots,y_n \in \Delta^I, y_i \neq y_j, 1 \leq i < j \leq n} t_{i=1}^{n}\{R^I(x, y_i)\}$

is derived from its first order formula

$\exists y_1, \ldots, y_n.\ \wedge_{i=1}^{n} R(x, y_i) \bigwedge \wedge_{1 \leq i < j \leq n} y_i \neq y_j.$

The semantics states that there are at least n distinct elements that satisfy to some degree.

Furthermore, since $\leq nR \equiv \neg(\geq (n+1)R)$, we define the semantics of a fuzzy at-most restriction as

$$\begin{aligned}(\leq nR)^I(x) &= \neg(\geq (n+1)R)^I(x) \\ &= \neg \sup_{y_1,\ldots,y_{n+1} \in \Delta^I, y_i \neq y_j, 1 \leq i < j \leq n+1} t_{i=1}^{n+1}\{R^I(x, y_i)\} \\ &= \inf_{y_1,\ldots,y_{n+1} \in \Delta^I, y_i \neq y_j, 1 \leq i < j \leq n+1} s_{i=1}^{n+1}\{\neg R^I(x, y_i)\}\end{aligned}$$

The FOL translation of a concept inclusion axiom $C \sqsubseteq D$ has the form $\forall x.C(x) \rightarrow D(x)$, therefore, its semantics is defined as

$(C \sqsubseteq D)^I(x) = \inf_{x \in \Delta^I} C^I(x) \rightarrow D^I(x) = \inf_{x \in \Delta^I}\{s(\neg C^I(x) \vee D^I(x))\}$.

Similarly, the semantics of a role inclusion axiom $R \sqsubseteq P$ is

$(R \sqsubseteq P)^I(x, y) = \inf_{x,y \in \Delta^I}\{s(\neg R^I(x, y) \vee P^I(x, y))\}$.

The semantics of the complex concept descriptions for $f\mathcal{ALCHIN}$ are summarized in Table 3.

Table 3: Syntax and Semantics of $\mathcal{ALCHIN}$ constructors

Constructor	Syntax	Semantics
top concept	$\top$	$\top^I = 1$
bottom concept	$\bot$	$\bot^I = 0$
atomic negation	$\neg A$	$(\neg A)^I(x) = \neg A^I(x)$
atomic negation	$\neg C$	$(\neg C)^I(x) = \neg C^I(x)$
concept conjunction	$C \sqcap D$	$(C \sqcap D)^I = t(C^I(x), D^I(x))$
concept disjunction	$C \sqcup D$	$(C \sqcup D)^I = s(C^I(x), D^I(x))$
exists restriction	$\exists R.C$	$(\exists R.C)^I(x) = \sup_{y \in \Delta^I}\{t(R^I(x, y), C^I(y))\}$
value restriction	$\forall R.C$	$(\forall R.C)^I(x) = \inf_{y \in \Delta^I}\{s(\neg R^I(x, y), C^I(y))\}$
inverse role	R^-	$(R^-)^I(y, x) = R^I(x, y)$
at-least restriction	$\geq nR$	$(\geq nR)^I(x) = \sup_{y_1,\ldots,y_n \in \Delta^I, y_i \neq y_j, 1 \leq i < j \leq n} t_{i=1}^{n}\{R^I(x, y_i)\}$
at-most restriction	$\leq nR$	$(\leq nR)^I x \equiv \neg(\geq (n+1)R)^I(x)$
concept inclusion axiom	$C \sqsubseteq D$	$(C \sqsubseteq D)^I(x) = \inf_{x \in \Delta^I}\{s(\neg C^I(x) \vee D^I(x))\}$
role inclusion axiom	$R \sqsubseteq P$	$(R \sqsubseteq P)^I(x, y) = \inf_{x,y \in \Delta^I}\{s(\neg R^I(x, y) \vee P^I(x, y))\}$
concept instance assertion	$C(a)$	$C^I(a)$
role instance assertion	$R(a, b)$	$R^I(a, b)$

4.3 Knowledge Bases in $f\mathcal{ALCHIN}$

A fuzzy knowledge base in $f\mathcal{ALCHIN}$ consists of a finite set of fuzzy axioms and fuzzy assertions. A fuzzy concept inclusion axiom has a form of $C \sqsubseteq D\ [l,u]\ (0 \leq l \leq u \leq 1)$ which describes that the subsumption degree between concept C and D is from l to u.

For example, the axiom

$Professor \sqsubseteq (\exists publishes.Journalpaper \sqcap \exists teaches.Graduatecourse)\ [0.8, 1]$

states that the concept professor is subsumed by publishing journal papers and teaching graduate courses with a certainty degree of at least 0.8.

A fuzzy role inclusion axiom has the form $R \sqsubseteq P\ [l,u]\ (0 \leq l \leq u \leq 1)$. A fuzzy concept assertion and a fuzzy role assertion are of the form $C(a)\ [l,u]$ and the form $R(a,b)\ [l,u]$ respectively.

5 Related Work

Uncertainty is known as an intrinsic feature of the World Wide World and Semantic Web. W3C even founded a group, the Uncertainty Reasoning for the World Wide Web (URW3) Incubator Group, which is dedicated to define the challenge of representing and reasoning with uncertain information. According to the latest URW3 draft report, uncertainty is a term intended to include different forms of incomplete knowledge, including incompleteness, inconclusiveness, vagueness, ambiguity, and others [8]. Mathematical theories for representing uncertainty information includes, but not limited to, probability, Fuzzy Sets, Belief Functions, Random Sets, Rough Sets, and combination of several models (Hybrid).

There has been some work carried out in integrating uncertainty knowledge into Description Logics in the last decades [6][7][12][13][14][11][9]. Current literature generally can be divided into two approaches. One is based on probabilistic theory [6][7][9] and the other is based on fuzzy logic [15][12][13][14][11]. Although both approaches assign numerical values to entries in a knowledge base, they are quite different; not only from a technical point of view, but also with respect to the basic phenomena they are trying to model. Probabilistic theory refers to a proposition that is either true or false, but due to a lack of information we do not know for certain which one is the case. It represents the probability with which a proposition is assumed to be true. For example, John can be assumed to be a student with the probability 0.6 and a teacher with the probability 0.4. On the other hand, fuzzy logic is used to represent the vagueness of a proposition, which means the proposition itself is only true to a certain degree. For example, John, measuring 1.85m, might be said to be tall with the degree of truth 0.9.

Our fuzzy description logic extended the expressiveness of the fuzzy ALC in [15][12] to support at-least and at-most number-restriction, as well as inverse-role and role-hierarchy constructors. Unlike other approaches based on fuzzy logic [15][12][13][11][16] which only deal with crisp subsumption of fuzzy concepts, our fuzzy description logic deals with fuzzy subsumptions of fuzzy concepts and addresses its semantics. We believe that fuzzy subsumption of fuzzy concepts in the form of is closer to the uncertain knowledge existing in the real world applications. [14] first proposed the notion of fuzzy subsumption but only used a form ,

while our approach generalizes it to a range of certainty values. Furthermore, we use general t-norm, s-norm, negation and implication in the semantics of our proposed fuzzy description logic, such that the interpretation of complex concept descriptions can follow different types of operations in Fuzzy Logic, such as, Zadeh Logic, Lukasiewicz Logic, Product Logic, and Gödel Logic.

6 Conclusion and Future Work

In this paper, we proposed an extension to Description Logics based on Fuzzy Set Theory and Fuzzy Logic. The syntax and semantics of the proposed description logic $f\mathcal{ALCHIN}$ were explained in details. We also addressed the components of a $f\mathcal{ALCHIN}$ knowledge base.

Description Logics is a family of description languages with different expressiveness. Our fuzzy description language extends the fuzzy $\mathcal{ALC}$ and takes into account inverse roles, role inclusion axioms, and number restrictions, but leaves alone transitive roles, nominals (i.e. collection of individuals) and datatypes for the reason of simplicity. Future work will include a fuzzy extension to more expressive description languages.

From the point view of reasoning with a $f\mathcal{ALCHIN}$ knowledge base, we present different reasoning tasks and the reasoning algorithm in another upcoming paper, because of the length limit here.

From the point view of implementing a corresponding reasoner, the plan is to build it on top of Pellet (http://clarkparsia.com/pellet/). Pellet is an open-source Java based OWL DL reasoner. Our extension of Pellet will provide functionalities to check consistency, entailments and subsumptions of a knowledge base.

References

[1] Baader, F., Calvanese, D., Mcguinness, D., Nardi, D., and Patel-Schneider, P. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, Cambridge, MA, 2003.

[2] Baader, F., and Sattler, U. An overview of tableau algorithms for description logic. *Studia Logica 69*, 1 (2001), 5–40.

[3] Berners-Lee, T., Hendler, J., and Lassila, O. The semantic web. *Scientific American 284*, 5 (2001), 34–44.

[4] Damasio, C. V., Pan, J., Stoilos, G., and Straccia, U. Representing uncertainty in ruleml. *Fundamenta Informaticae 82* (2008), 1–24.

[5] Hjek, P. *Metamathematics of fuzzy logic*. Kluwer, 1998.

[6] Jaeger, M. Probabilistic reasoning in terminological logics. In *Proc. of the 4th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR94)* (1994), pp. 305–316.

[7] Koller, D., Levy, A., and Pfeffer, A. P-classic: A tractable probabilistic description logic. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)* (1997), pp. 390–397.

[8] Laskey, K., Laskey, K., Costa, P., Kokar, M., Martin, T., and Lukasiewicz, T. W3c incubator group report. Tech. Rep. http://www.w3.org/2005/Incubator/urw3/wiki/DraftFinalReport, W3C, 05 March, 2008.

[9] Lukasiewicz, T. Expressive probabilistic description logics. *Artificial Intelligence 172*, 6/7 (2008), 852–883.

[10] Novk, V. *Mathematical principles of fuzzy logic*. Dodrecht: Kluwer Academic, 1999.

[11] Stoilos, G., Stamou, G., Pan, J., Tzouvaras, V., and Horrocks, I. Reasoning with very expressive fuzzy description logics. *Journal of Artificial Intelligence Research 30* (2007), 273–320.

[12] Straccia, U. A fuzzy description logic. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI'98)* (1998), pp. 594–599.

[13] Straccia, U. Reasoning within fuzzy description logics. *Journal of Artificial Intelligence Research 14* (2001), 137–166.

[14] Straccia, U. Towards a fuzzy description logic for the semantic web (preliminary report). In *2nd European Semantic Web Conference (ESWC-05)* (2005), Lecture Notes in Computer Science, Springer Verlag, pp. 167–181.

[15] Tresp, C. B., and Molitor, R. A description logic for vague knowledge. In *Proc. of the 13th Eur. Conf. on Artificial Intelligence (ECAI'98)* (1998), pp. 361–365.

[16] Venetis, T., Stoilos, G., Stamou, G., and Kollias, S. f-dlps: Extending description logic programs with fuzzy sets and fuzzy logic. In *Fuzzy Systems Conference, 2007. FUZZ-IEEE 2007. IEEE International* (2007), pp. 1–6. ID: 1.

[17] Zadeh, L. A. Fuzzy sets. *Information and Control 8*, 3 (1965), 338–353.

Generating partial COP-nets on demand

Henry Bediako-Asare[1], Michael Fleming[1], Scott Buffett[2]

1.Faculty of Computer Science, University of New Brunswick *2. National Research Council of Canada*

Background Information

The idea of agents representing users in negotiations has encouraged research in preference elicitation and user preference modeling. The Conditional Outcome Preference Network (COP-net) is used to predict preferences over an entire set of possible outcomes, given a (typically very small) number of elicited preferences.

Problem Definition

The existing methodology uses **all** possible outcomes to construct a COP-net. Thus for very large numbers of outcomes, the construction of the COP-net becomes infeasible. The table below gives an idea of the exponential growth of possible outcomes.

Number of attributes	Number of values per attribute	Number of outcomes generated for constructing the COP-net
5	4	1,024
15	4	1,073,741,824
15	6	470,184,984,576
20	9	12,157,665,459,056,928,801

Proposed Solution

Using A* Search, a **partial** COP-net involving only some of the outcomes that are relevant in making a prediction, instead of one with all possible outcomes, is constructed.

Methodology

Four paths are built using A* Search. They are then merged to obtain the partial COP-net, as shown in the following example.

Six attributes of a BMW 7 series, each having two values, resulting in 64 possible outcomes

Colour = {white, silver}
Engine = {V8, V10}
Drive train = {FWD, RWD}
Wheels = {18" star, 19" star}
Interior = {brown leather, black leather}
Package = {premium, sport}

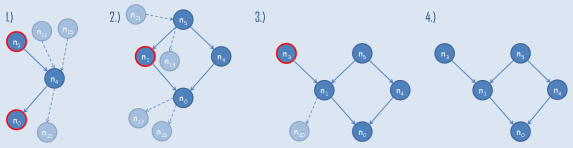
Elicited preferences from a user

silver > white; V8 > V10; FWD > RWD; 18" star > 19" star; brown leather > black leather; premium > sport

Suppose we want to find out if a **BMW 7 series** which is white in Colour, has a **V8** Engine, is Rear Wheel Drive, has **19"** star shaped alloy Wheels, has a black leather Interior and has a sports Package (**wV8R19blsp** or n_3) is preferred over one that is silver in Colour, has a **V10** Engine, is Front Wheel Drive, has **19"** star shaped alloy Wheels, has a black leather Interior and has a sports Package (**sV10F19blsp** or n_4).

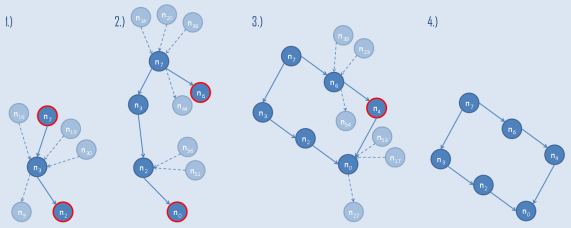
Nodes	n_0	n_1	n_2	n_3	n_4	n_5	n_6	n_7 ...
Outcomes	sV10R19blsp	wV10R19blsp	sV8R19blsp	wV8R19blsp	sV10F19blsp	wV10F19blsp	sV8F19blsp	wV8F19blsp ...

Nodes created for the 8 outcomes involved in the partial COP-net

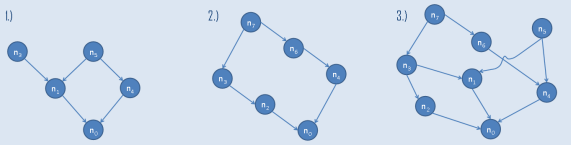


First pair of paths generated from n_4 to n_3

Methodology (continued)



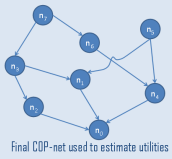
Second pair of paths generated from n_3 to n_4



First pair of paths | Second pair of paths | Resulting partial COP-net

The first and second pairs of paths are merged to obtain a partial COP-net involving 8 outcomes instead of all 64 possible outcomes

Utilities – quantitative measures of how desirable an outcome is – are estimated for the outcomes in the partial COP-net, using an existing technique. In this case, since the estimated utility for n_3 is greater than that for n_4, it is predicted that the outcome represented by n_3 is preferred over the outcome represented by n_4.



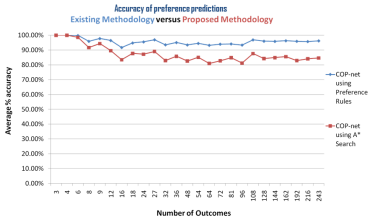
Final COP-net used to estimate utilities

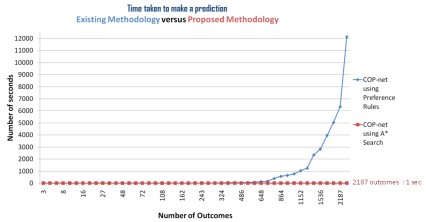
Nodes	n_0	n_1	n_2	n_3	n_4	n_5	n_6	n_7
Estimated Utilities	0.0	0.333	0.333	0.666	0.333	1.0	0.666	1.0

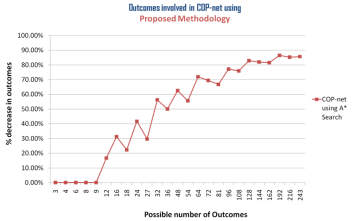
Estimated utilities for the nodes in the partial COP-net

Current Results

The charts below show the current results of how the existing methodology compares with the proposed methodology in terms of the average accuracy in predicting preferences of outcomes, the time it takes to make one prediction, and the number of outcomes involved in both methodologies' COP-nets.







Faculty of Computer Science UNB

National Research Council Canada | Conseil national de recherches Canada

Faculty of Computer Science UNB

A Framework for Mobile Applications

Sangwhan Cha

Supervisors : Dr. Weichang Du
Dr. Bernd J. Kurz

Introduction

Mobile application developers and content providers usually need to develop mobile applications with concerns for mobility for specific wireless networks and device platforms which are used by network carriers.

In order to provide standard mobile applications with interoperability and mobility support, we propose a comprehensive mobile application framework to support interoperability and mobility of mobile application development and operation.

Such framework supports developing mobile device applications, mobile server applications, as well as mobile client-server communications and peer-to-peer communications

Research Problem

Interoperability ?

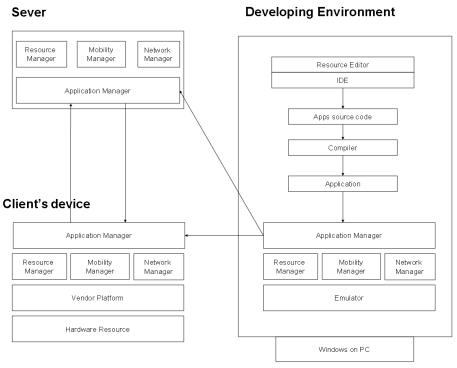
: The ability of applications to be executable across diverse mobile platforms

Mobility?

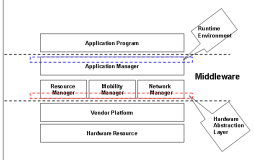
:The ability of applications to be seamlessly accessible across heterogeneous networks and devices

"No known framework for supporting mobile applications with both interoperability and mobility"

Architecture Overview

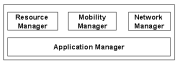


Client Architecture



- Application Manager : installation, deletion, and execution of applications
  runtime Environment, basic APIs and Components
- Resource Manager : limited device resources and capabilities
  device-related APIs and components
- Mobility Manager : location awareness
  location-related APIs and mobility-related APIs
- Network Manager : wireless connectivity
  network-related APIs

Server Platform



Functionality of server platform

- Context handling
  Network manager : Network context handling
  Resource manager : Device context handling
- Content adaptation
  Application manager
- Automatic service selection
  Mobility manager
- Provision of services based on contextual information
  Application and content provider in Application manager

Future Work

Developing Middleware for Devices

The test bed for the prototype of the unified-middleware platform

Evaluation points:

- Interoperability:
  - Runtime environment
  - Hardware Abstraction Layer
- Mobility structured platform

Developing the Server Platform

The test bed for the prototype of the server platform

Evaluation points :

- Context handling
- Automatic service selection
- Provision of services based on contextual information
- Multimedia content handling while the user is moving

Developing the Developing Environment

The test bed for the prototype of the unified developing environment

Evaluation points:

- Supporting multi programming languages (C & Java)
- Supporting mobility specified APIs

General evaluation points:

- Delay of the service provision for each specific network and device
- Delay of the service provision when vertical handoff occurs

Reference

- For more information, Refer to the paper : S. Cha, B. Kurz, and W. Du, "Toward a unified middleware for mobile applications" in Communication Networks and Research Conference (CNSR 2009) IEEE, Accepted.

Communication Networks and Services Research (CNSR)

This research is supported and funded through CNSR by Bell / Aliant and ACOA by an AIF research contract.

Update Propagation In Modular Ontologies

Faezeh Ensan and Weichang Du

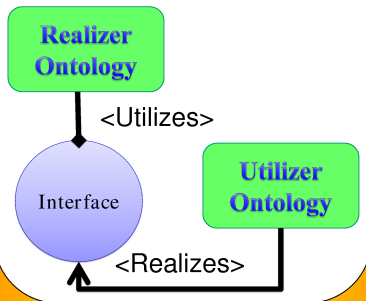
Introduction

- Ontology:
  - main building block of the Semantic Web .
- Modular ontology formalisms:
  - Provide basis for defining ontologies as a set of self-contained modules.
- Revisions and updates in ontologies
  - A modification may give rise to inconsistencies that should be resolved.

Contributions

- Analyzing the notion of updates in modular ontologies
- A method for TBox update propagation in modular ontologies

IBF: Interface Based Formalism for Modular Ontologies



TBox Modification

Utilizer knowledge base:

$A \sqsubseteq I{:}D$

$I{:}D \sqsubseteq B$

$B \sqsubseteq \neg I{:}C$

Modification in Realizer module:

$I{:}D \sqsubseteq I{:}C$

D is unsatisfiable in the utilizer module

ABox Modification

Utilizer knowledge base:

$A \sqsubseteq I{:}D$

$I{:}D \sqsubseteq B$

$B \sqsubseteq \neg I{:}C$

Modification in realizer module: **D(a), C(a)->** Inconsistent utilizer module

Resolving Inconsistencies

- Principle of zero change for internal elements .
- Principle of minimal change for external elements.
- External diagnosis for modular ontologies
- Removing the minimal external diagnosis and producing a consistent integrated knowledge base.

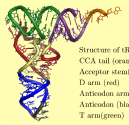
Faculty of Computer Science, University of New Brunswick

RNA Motif Discovery using Probabilistic Tree Adjoining Grammars

Patricia A. Evans and Emad Bahrami Samani {pevans,emad.b.samani}@unb.ca University of New Brunswick

Introduction

RNA is an informational molecule which plays an important role in living organisms. RNA not only is the main component in transcription in cell and protein construction but also its 3D structure allows it to be a biocatalyst. Nucleic acid targeted drug design which mainly takes advantage of the RNA in the cell is an strong hope to cure huge trouble-making diseases such as AIDS and cancer. Finding patterns in RNA structures is the first step in this way. Thus, RNA structural motif discovery has immediate applications in medicine.



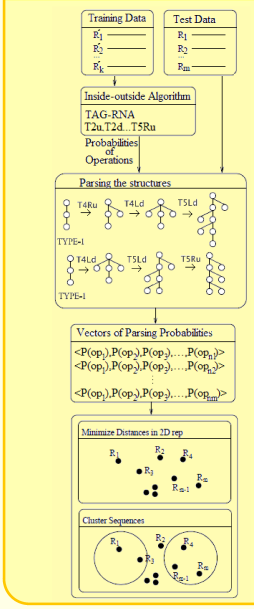
Structure of tRNA - CCA tail (orange), Acceptor stem(purple), D arm (red) Anticodon arm (blue) Anticodon (black), T arm(green)

This problem is a difficult one. Firstly because we need to deal with large amounts of biological data to recognize complex patterns. Secondly, we need methods to direct biological experiments. Pseudoknots are very important types of structure elements in RNA but it has been proven that modeling the RNA secondary structure with arbitary pseudoknots is NP-Complete. There are several methods in the literature trying to tackle this problem but a fast and accurate method that copes with pseudoknots seems necessary. This project proposes a new technique to extract the structural motifs of RNA molecules using Tree Adjoining Grammars. The main advantage of our method is that it can efficiently model pseudoknots in RNA secondary structures and extract motifs containing these structures accurately and fast. There have also been several grammatical approaches to modeling some kinds of pseudoknots. In the grammatical approaches, secondary structure modeling can be done during the process of the parsing the grammars, which can be addressed in $O(n^4)$ to $O(n^6)$ time. "Tree Adjoining Grammars" has been proposed as a useful formalism for the study of natural languages. A tree-adjoining grammar(TAG) contains two sets of elementary structures: initial trees and auxiliary trees. These elementary structures can be combined using two operations, substitution and adjunction [1].

Probabilistic TAGs

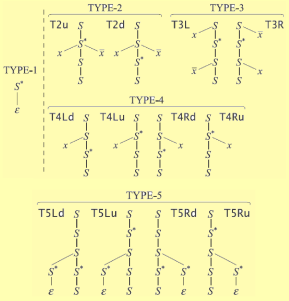
A probabilistic tree-adjoining grammar is a 5-tuple, (I, A, P_I, P_S, P_A), where I and A are Initial and Auxilary trees defined as above, P_I is a function that is known as the probability that a derivation begins with an initial tree. P_S is the probability of substitution operation and P_A denotes the probability of adjoining operations [3].

Proposed Solution in a glance



Modeling Pseudoknots

Tree adjoining grammars are described as mildly context-sensitive as they possess certain properties that make them more powerful than context-free grammars, but less powerful than context-sensitive grammars[2, 3]. Mild context sensitivity is useful for defining dependency between different parts of the string generated by the grammar. Therefore, it can be used to model pseudoknotted RNA secondary structures. In order to handle the secondary structures of RNA including pseudoknots, we will develop a novel algorithm to use the Probabilistic Tree Adjoining Grammars for RNA, denoted by TAG-RNA [2]. We will derive the TAG tree of each sequence and its annotated structure using TAG-RNA. Given an RNA sequence with its annotation of secondary structure including pseudoknots, a TAG derivation is obtained by parsing the RNA secondary structure with TAG-RNA.



Our parsing algorithm is based on the algorithm by Vijay-Shankar ([4]). TAG-RNA parser is a bottom-up parsing algorithm in nature. It uses four-dimensional dynamic programming method and can find an optimum solution with respect to some evaluation functions. The time and space complexity is $O(n^4)$, where n is the length of an input string. We use a TAG derivation process to find the common motifs between the secondary structure of two RNA [5].

Extracting Motifs

According to [3], if two points x and y, have probabilities P(x) and P(y), then their mutual information, I(x,y), is defined to be:

$$I(x,y) = \log_2 \frac{P(x,y)}{P(x)P(y)} \quad (1)$$

Mutual information compares the probability of observing x and y together with the probabilities of observing x and y independently. If there is a genuine association between x and y, then P(x,y) will be much larger than the probability P(x)P(y), and consequently $I(x,y) \gg 0$. If there is not any relationship that would be interesting for us between x and y, then P(x,y) $\simeq$ P(x)P(y), and so, $I(x,y) \simeq 0$. If x and y are in complementary distributions, then P(x,y) will be much less than P(x)P(y), forcing $I(x,y) \ll 0$.

To find novel RNA structural motifs we will calculate the mutual information between every two adjacent operation in derivation $\tau = (\alpha_0, op_1(..), op_2(..),..., op_n(..))$. Then we will develop a dynamic programming algorithm to find the similar patterns in these vectors. A set H of points in Euclidean space is selected so that for each sequence $s \in G$ there is a corresponding point $P(s) \in H$. Principle Coordinates Analysis (PCoA) will be used to find corresponding points in 2D space [7]. The points in H are examined to find clusters. PCoA automatically projects to the subspace where the global solution of K-means lies. RNA structural motifs are the different clusters.

References

[1] K.V. Shanker, D.J. Weir and A. K. Joshi, "Characterizing structural descriptions produced by various grammatical formalisms", *25th Annual Meeting of the Association for Computational Linguistics (ACL)*, 1987.

[2] Matsui, H., et. al., "Pair stochastic tree adjoining grammars for aligning and predicting pseudoknot RNA structures", *Bioinformatics, 21(11):2611-2617, 2005.*

[3] Church, K. W. and Hanks, P., "Word Association Norms, Mutual Information, and Lexicography", *Computational Linguistics, 16(1):22-29, 1990.*

[4] Vijay-Shankar, k. and Joshi, A. K., "Computational Properties of Tree Adjoining Grammars", *ACL*, 1985.

[5] Uemura, Y. et. al., "Grammatically Modeling and Predicting RNA Secondary Structures", *Proc. Genome Informatics Workshop, 1995.*

[6] J. Schonfeld and D.A. Ashlock, "Evaluating Distance Measures for RNA Motif Search", *Congress on Evolutionary Computation, 2006.*

[7] *http : //2008.igem.org/wiki/images/3/3f/TRNA.png*

I/O Efficient Search of Moving Objects on a Graph

Thuy T. T. Le and Bradford G. Nickerson

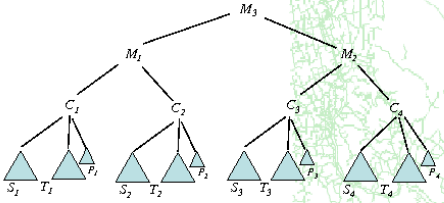
Faculty of Computer science, University of New Brunswick, Fredericton, New Brunswick, Canada

Motivation

- Queries on historical positions of moving objects on planar graphs are likely to be used in applications such as reconstruction, planning, simulation and training.
- How to efficiently store the position history of moving objects?
- How to improve the response time (disk accesses) for query processing of moving objects?
- Define: Graph G with n moving object instances on E edges.

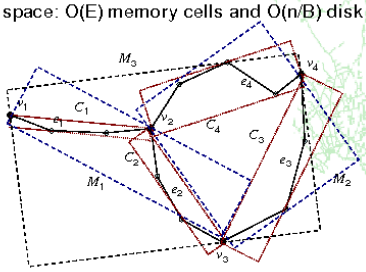
Data Structure

- Minimum I/O Graph strip tree (minGStree)

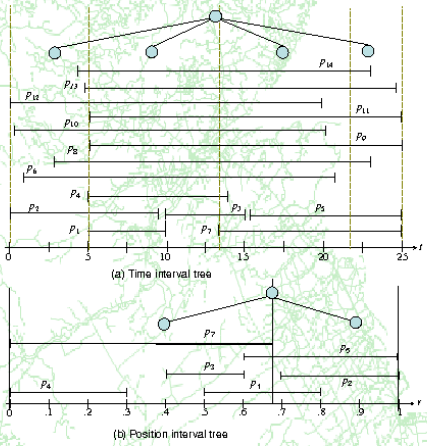


Each leaf C_i points to the strip tree S_i representing edge e_i, and two corresponding interval trees: time interval tree T_i and position interval tree P_i. Interval trees T_i are kept in external memory.

- Required space: O(E) memory cells and O(n/B) disk blocks.



Four edges e_1, ..., e_4 as four strip trees whose bounding boxes are C_1, ..., C_4. Strip trees are merged bottom up in pairs to construct the minGStree.



(a) Time interval tree

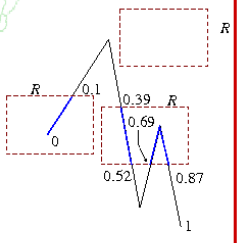
(b) Position interval tree

Obj. id	Time Int.	Position Int.
p1	5 10	0.5 0.8
p2	0 9	0.7 1
p3	10 15	0.4 0.6
p4	5 13	0 0.3
p5	16 25	0.6 1
p6	2 22	0 1
P7	12 25	0 0.68
p8	3 23	0 1
p9	5 25	0 1
p10	1 21	0 1
p11	5 25	0 1
p12	0 20	0 1
p13	4 24	0 1
p14	3 23	0 1

An example of 14 object instances on an edge. Their 14 time intervals are indexed in T_i. The six non-[0,1] position intervals are indexed in P_i

Searching

- Query $Q_2 = (R, [t_1,t_2])$, R is a rectangular query, $[t_1, t_2]$ is a query time interval. Q_1 is Q_2 with $t_1 = t_2$.
- Starting at the root node, find edges e_i intersecting R by strip tree rectangle search on appropriate leaf nodes C_i.
- For each intersected edge e_i, time interval tree T_i is used to find moving objects satisfying the time interval query $[t_1, t_2]$, and then P_i is used to prune moving objects not intersecting R.



- Q_1 and Q_2 queries require $O(\log_B n/E + k)$ disk I/Os, where k is the number of disk blocks required to store the answer; B is the number of objects transmitted by one disk I/O.

Sponsored by:

madalgo

UNB Computer Science

Harrison McCain Foundation

NSERC CRSNG

Effective Query Selection during Preference Elicitation

Minruo Li[1], Michael Fleming[1], Scott Buffett[2]

1. Faculty of Computer Science, University of New Brunswick

2. National Research Council of Canada

Objective

One of the problems in conducting automated negotiation is that of maximizing the utility of the user being represented. However, the agent conducting the negotiation typically will not have prior knowledge of the user's true utility for every outcome. This project tries to find an efficient and accurate way for the agent to estimate the user's utility for each outcome and use this estimated utility when making decisions during the negotiation process.

Background – COP-nets

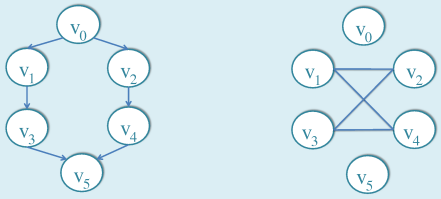
A Conditional Outcome Preference Network (COP-net) is a graphical model used to represent a user's preferences over a set of outcomes [1].

- A COP-net is a directed acyclic graph consisting of a set of nodes and a set of directed edges.
- Each node denotes an outcome and each directed edge represents a preference.
- COP-nets are transitively reduced graphs.
- A COP-net has a small number of nodes with a prior labeling of known utilities for the user, which are known as true utilities and will then be used, along with the known preferences, to estimate the utilities of the rest of the nodes for the user.

Query Selection with COP-nets

Our goal is to find some currently *unknown* preferences that would be the most useful to the agent. This can be modeled with the COP-net as well. The procedure for finding all unknown preferences using a COP-net is summarized as follows:

- Find the transitive closure C_{TC} of a COP-net
- "Undirect" the graph, by converting arcs to undirected edges
- Find the complement C'_{TC}, which is referred to as the query graph
- Each edge in the resulting graph represents an unknown preference or a query



Methodology

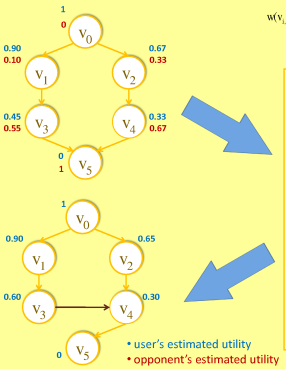
1. Construct the COP-net
2. Estimate user's utility of each outcome using an existing method (longest path method)
3. Generate query graph and find all possible queries (unknown preferences)
4. **Weight each query**
5. Ask one or more queries with highest weight
6. Re-construct the COP-net and re-estimate user's utility of each outcome
7. **Simulate the negotiation process**

Weighting Scheme

We would like to ask queries that reveal a high number of unknown preferences and that involve outcomes that are more likely to be interesting for both the user and the opponent. In one proposed method, the steps for calculating the weight for each edge in the query graph are described as follows:

- Let o_i be the outcome represented by v_i in the query graph
- Let (v_i, v_j) be an edge in the query graph
- Let P_{oij} be the set of preferences that would be learned if the user specifies $o_i > o_j$, including $o_i > o_j$
- Let E_{oij} represent the set of edges that would be removed from the query graph as a result
- Let $u(o_i)$ and $u_{opp}(o_i)$ be the agent's estimates of the user's and the opponent's utilities for o_i
- Compute weight of (v_i, v_j) by the formula:

$$w(v_i, v_j) = \min\left\{ \sum_{(v_k, v_l) \in E_{oij}} (u(v_k) \times u_{opp}(v_k) \times u(v_l) \times u_{opp}(v_l)), \sum_{(v_k, v_l) \in E_{oji}} (u(v_k) \times u_{opp}(v_k) \times u(v_l) \times u_{opp}(v_l)) \right\}$$



Weight(v_1, v_2) = min{0.90*0.10*0.67*0.33
+ 0.90*0.10*0.33*0.67,
0.67*0.33*0.90*0.10
+ 0.67*0.33*0.45*0.55}
= 0.0398

Weight(v_1, v_4) = min{0.90*0.10*0.33*0.67,
0.33*0.67*0.90*0.10
+ 0.33*0.67*0.45*0.55}
= 0.0199

Weight(v_3, v_4) = min{0.45*0.55*0.33*0.67
+ 0.90*0.10*0.33*0.67,
0.33*0.67*0.45*0.55
+ 0.67*0.33*0.45*0.55}
= **0.0746**

Weight(v_2, v_3) = min{0.67*0.33*0.45*0.55,
0.45*0.55*0.67*0.33
+ 0.45*0.55*0.33*0.67}
= 0.0547

Testing – Negotiation process

Both the user and the opponent will give an offer that maximizes their own utilities and accept an offer only when their utility for the offer reaches some acceptance point.

Outcomes	User's True Utility	User's Estimated Utility: Without asking queries	User's Estimated Utility: After asking queries	Opponent's True Utility
v_0	1	1	1	0
v_1	0.85	0.90	0.80	0.20
v_2	0.50	0.60	0.55	0.40
v_3	0.30	0.50	0.25	0.70
v_4	0.20	0.40	0.15	0.85
v_5	0	0	0	1

(Accepting point = 0.40)

Negotiation without asking queries:
- The user offers v_0
- The opponent rejects and offers v_5
- The user rejects and offers v_1
- The opponents rejects and offers v_4
- The user accepts v_4

Final utility achieved: 0.20

Negotiation with asking queries:
- The user offers v_0
- The opponent rejects and offers v_5
- The user rejects and offers v_1
- The opponents rejects and offers v_4
- The user rejects and offers v_2
- The opponent accepts v_2

Final utility achieved: 0.50

The negotiation process will be conducted with different weighting schemes. At the end of each negotiation process, we will measure the user's true utility of the accepted offer and compare the results obtained from the use of several different query weighting methods.

References

[1] S. Chen, S. Buffett, and M. W. Fleming. Reasoning with Conditional Preferences across Attributes. 2007. The 20th Canadian Conference on Artificial Intelligence May 28, 2007.

Faculty of Computer Science UNB

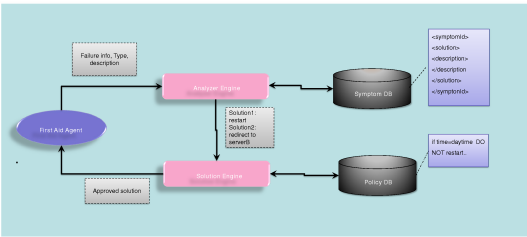
National Research Council Canada | Conseil national de recherches Canada

Self-Healing Power Grid by Autonomous Agent Framework

Zeinab Noorian, Hadi Hosseini,
Faculty of Computer Science, UNB Fredericton

Introduction

We have proposed an adaptive self-healing framework for agent-based power grids. It endows power grid with self-awareness such that it is able to identify emerging vulnerabilities in order to reconfigure itself to attain resilience for different types of failures. Furthermore, it provides a cognitive planning cycle to find ultimate corrective solution as well as evaluation service to verify the effectiveness and performance of the final solution.



How the Self-Healing Framework Works?

❑Sensor agents report malfunctioning like reduction in wind power to Observer Agent Manager.

❑OAM validates the report by accessing Information Server. It triggers the Failure Detection Agent.

❑Failure Detection agent examine the report then splits it up and dispatch them to the corresponding agents:

❑Quarantine&Isolation agent, uses isolation mechanism to prevent propagation of the failure throughout the grid.

❑First Aid agent, invokes Autonomic Web Services to find a temporary self-healing solution.

❑Event Identification agent decodes failure description and sends this information to the Vulnerability Assessment agent in order to assess the criticality of the damage. Afterwards, it directs this information to the Planning agent.

❑Planning agent access History Knowledge base then, offer various recovery solutions to Decision-Making agent

❑Decision Making agent finalize the optimized solution according to predefined system factors such as robustness, reliability and security and then forward the result to Reconfiguration agent.

❑Reconfiguration agent attaches the system adaptation requirements to the ultimate recovery solution then send it to Evaluation agent.

❑Evaluation agent measure up the effectiveness of the solution and if it satisfies system threshold, it will execute the solution on infected area.

Self-Healing Framework for Power Grid

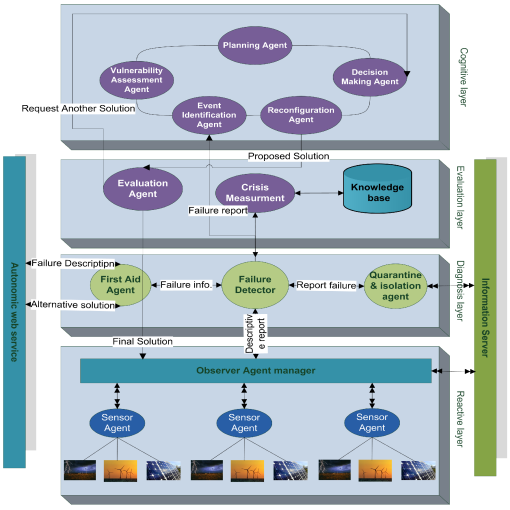
The proposed self-healing framework contains four hierarchical layers:

I. *Reactive Layer* provides monitoring services which notify slight changes in power.

II. *Diagnosis Layer* provides failure detection service and also it takes advantage of Autonomic Web Services in determining the temporary corrective solutions as a first-aid action in order to prevent system from blackout even for a short time.

III. *Cognitive Layer* is designed for identifying multiple solutions and examining the effectiveness of them toward system configuration and policy.

IV. *Evaluation Layer* consists of evaluation and crisis measurement agents to finalize the optimized solution and send the request for execution.



Conclusions

The agent-based self-healing power grid is aware of emerging troubles and is able to reconfigure itself to resolve the problems and could reduce blackouts dramatically. It can sense local problems at early stages, and automatically fix or isolate them before they grow larger; this is needed to prevent the cascading power failures that cause blackouts

ADAPTIVE RISK MANAGEMENT LABORATORY

Prof. Mihaela Ulieru
Canada Research Chair
Director ARM Laboratory

UNIVERSITY OF NEW BRUNSWICK

UNB

Performance Enhancement of Smith-Waterman Sequence Database Searches Using Hybrid Model: Comparing the MPI and Hybrid Programming Paradigm on SMP Clusters

Mahdi Noorian and Zeinab Noorian
Faculty of Computer Science, UNB Fredericton

1. Introduction

Nowadays, database pattern searching is the most heavily used operation in computational biology. Indeed, sequence alignment algorithm plays an important role to find the homologous groups of sequences which may help to determine the function of new sequences. Meanwhile Smith-Waterman algorithm is one of the most prominent pattern matching algorithms. However, it cost the large quantity of time and resource power. By the aid of parallel hardware and software architecture it becomes more feasible to get the fast and accurate result in efficient time. Here we intent to show that the hybrid programming which employ the coarse grain and fine grain parallelization, is more efficient compare with pure MPI and pure OpenMP in cluster of SMP machines.

2. Hybrid; MPI + OpenMP

The hybrid programming model provides opportunity to take advantage of the both MPI and OpenMP models at the same time. The hybrid programming model instinctively matches with the structural characteristics of a cluster of Shared Memory Processors(SMP) nodes, as well as providing two level communication patterns: intra and inter-node communication . Intra-node communication is feasible by the aid of OpenMP and thread programming model. Subsequently, inter-node communication is achieved through message passing Interface technology between nodes.

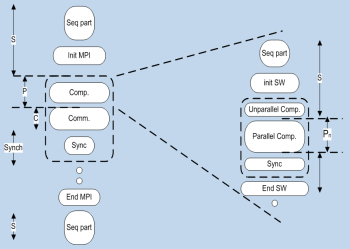


Fig.1 Hybrid Parallelizing by using MPI and OpenMP approach.

3. Smith-Waterman Algorithm

The Smith-Waterman algorithm is perhaps the most widely-used local similarity algorithm for biological sequence pair wise alignment. The algorithm consists of three steps:

1.Fill the dynamic programming matrix.
2.Find the maximal value in the matrix.
3.Trace back the path that leads to maximal score to find the optimal local alignment.

$$R_{ij} = \max \begin{cases} R_{(i-1)(j-1)} + Sbt(s,t) \\ R_{(i-1)j} + gap \\ R_{i(j-1)} + gap \\ 0 \end{cases}$$

4. Experimental Result

Based on abovementioned methodology, Smith-Waterman algorithm has been implemented using MPI and OpenMP and Hybrid paradigms. The performance of this parallel implementation was evaluated using various database sizes and a against specific query sequences.

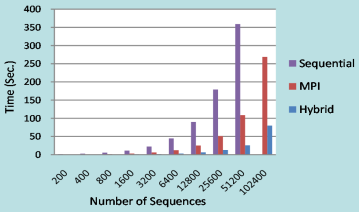


Fig.2 Comparison between three methods for different data files with six CPUs

As it is depicted in Fig 2, the hybrid model which get benefit from both MPI and OpenMP technology, obtain better result and performance in terms of execution time compare with pure MPI parallel implementation and clearly sequential model. Note that, the sequential implementation of the algorithm can not be executed for 102400 numbers of sequences.

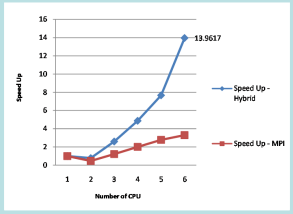


Fig. 3 Comparison speed up between MPI and Hybrid methods for data file with 3200 Sequences

In addition, Fig 3 demonstrates speed up of MPI and Hybrid model for a specific input data across different processor number.

5. Conclusion

The hybrid implementation of the Smith-Waterman algorithm is presented, which combines fine grain and coarse grain parallelism and multi-level scheduling. This implementation achieved a speed up fourteen on a cluster five Quad-Core Intel Xeon 1.6GHz as workers and one as master.

Nowadays, it becomes an obligation to use hybrid implementation by taking advantages from multi-core processors technology in clusters of SMP machine. Since a processor in a cluster contains of different cores and each core can run a thread separately, these types of clusters can give a significant speed up with hybrid implementation as compared with pure MPI. Breaking job in to the threads at the processor level and using share memory with high speed bus connections gives us the opportunity to decrease the execution time in hybrid model.

UNIVERSITY OF NEW BRUNSWICK UNB

UNB Honeynet environment

Hanli Ren, Natalia Stakhanova, Wei Lu, Ali Ghorbani

UNB honeynet is a member of Canadian honeynet chapter (http://honeynetproject.ca) that is an official chapter of Honeynet Project, a global organization spawning honeynet chapters of all countries

A honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource.

A honeynet is a high-interaction honeypot designed to capture extensive information on threats.

Low-interaction honeypot

Solution emulates operating systems and services.
- Easy to install and deploy. Usually requires simply installing and configuring software on a computer.
- Minimal risk, as the emulated services control what attackers can and cannot do.
- Captures limited amounts of information, mainly transactional data and some limited interaction.

High-interaction honeypot

No emulation, real operating systems and services are provided.
- Can capture far more information, including new tools, communications, or attacker keystrokes.
- Can be complex to install or deploy (commercial versions tend to be much simpler).
- Increased risk, as attackers are provided real operating systems to interact with

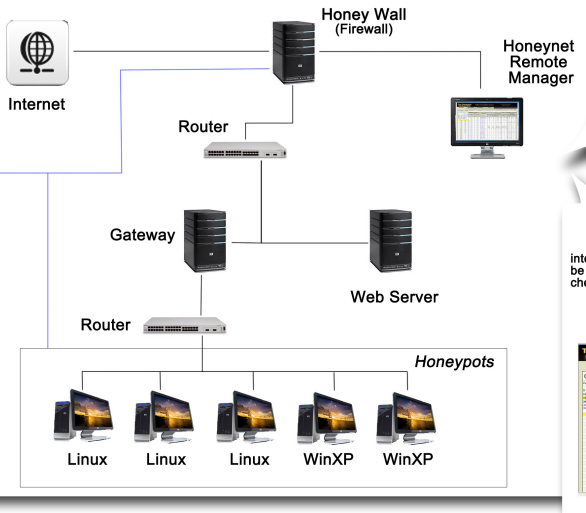
Gen III Honeynet framework

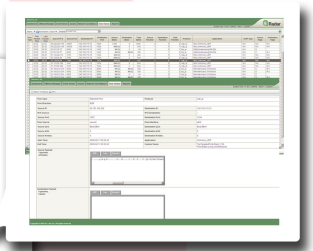
supports High-Interaction Honeypots

Sebek

a data capture tool designed to capture attacker's activities on a honeypot, without the attacker (hopefully) knowing it.

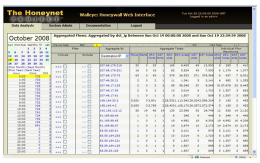
Honeynet Architecture & Technology we use





Walleye

interface to administering the Honeywall, can be also used for full system data analysis, e.g., check network flows, Snort alerts.



Why do we need honeynet?

- Data capture
  - Analysis of the security threats and vulnerabilities
  - Investigate tactics and practices of hacker community
- New tools deployment
  - Evaluation of detection & response effectiveness
- Students' education

Attacks we see

- SQL slammer worm
- FTP brute force
- Web server scans

Faculty of Computer Science
University of New Brunswick

Botnet Analysis Framework

Ali Shiravi and Ali A. Ghorbani {ali.shiravi, ghorbani}@unb.ca

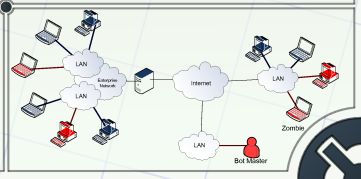
Introduction

Botnet is a coordinated group of malware instances (bots) that are controlled by a botmaster via some command and control (C&C) channel.

Botnets are the largest threat to Internet security. Most of the attacks and fraudulent activities on the Internet are carried out by malicious software, i.e., malware, which includes viruses, trojan, worms, spyware, and recently botnets. Such malware has risen to become a primary source of most of the scanning, distributed denial-of-service (DDoS) activities, direct attacks, and fraudulent activities taking place across the Internet.

All bots distinguish themselves from the previous malware forms by their ability to establish a command and control (C&C) channel through which they can be updated and directed by a botmaster. Traditionally 3 structures are defined for botnets:

1) Centralized
2) Peer-toPeer
3) Random



Background

Research community has been pursuing techniques to detect and respond to these malicious activities. A variety of techniques have been developed and applied to various data sources. Counteracting this emerging threat requires better techniques that assist in identify botnets (bots and/or their C&C servers) and providing the mechanisms necessary to mitigate their damage and defend against them. Botnets exhibit properties that require new approaches for detection and elimination.

However, these projects have largely been pursued in isolation and final result as single-purpose collection of methods, analysis results, and response directives.

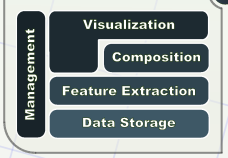
Much of the efforts within current projects are directed at tuning the system for a very specific C&C structure and bot behavior. As a result, systems tend not to provide comprehensive detection coverage for botnet activity across a spectrum of structures and communication protocols. Thus, to effectively cover the potential threats, it is desirable to provide a platform which integrates many detection systems that provide complementary coverage and provide the necessary mechanisms to customize the methods or manipulate their workings. This would necessarily mean the requirement to add, remove, rearrange, and rework the components in question.

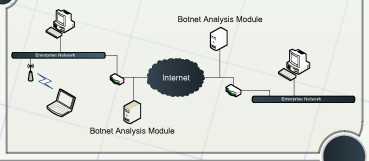
Objectives

We intend to addresses the research questions involved in implementing such a framework.

A Botnet Analysis Framework is a framework in which it defines, how to organize the modules, methods, components and structures associated with the concept of analysing Botnets, to gain better iinsight to how we should defend againt it.

The concept of defence here encompasses detection, mitigation, and visualization. The various top-level components are shown in the following diagrams.





Benefits

The design of this framework aims to facilitate botnet analysis by allowing enterprises, designers, researchers and programmers to spend more time on meeting requirements rather than dealing with the more standard low-level details of providing a working system and methods.

For example, an enterprise using such a framework to analyse the existance of bots on their network can focus on the operations of high level detectors and mitigation methods, rather than the mechanics of components facilitating defence.

An incomplete list of the overall features would be of the following attributes:

- Low learning curve to use the framework.
- Extensibility
- Scalability
- Open interfaces (API)

network security laboratory

UNIVERSITY OF NEW BRUNSWICK

UNB

Network Security Simulation Visualization

Ali Shiravi, Hadi Shiravi, and Ali A. Ghorbani {ali.shiravi, hadi.shiravi, ghorbani}@unb.ca

Introduction

Information visualization has matured over the past years and has been applied to a variety of applications. It has been only few years that this work been applied to the network security domain and other related information assurance problems.

Measuring the amount of damage caused by an attack is of great value. From this information network administrators can respond to these threats and ultimately protect the organizations data integrity and confidentiality.

Modeling and simulation is one of the corner stone's of Computer Science and its applications are widely acknowledged in network security evaluation. Here at NSL, we have developed an assessment tool to analyze the state of the network, in relation to different attack scenarios. The development of such security simulations and modeling tools present an interesting challenge which motivates the further research of visualizing information obtained from simulation.

Background

The basic idea with focus–plus–context–visualizations is to enable viewers to see the object of primary interest presented in full detail while at the same time getting a overview–impression of all the surrounding information — or context — available.

The currently existing focus-and-context methods are:

- Spatial methods. The image created with an existing visualization is distorted to allow more space for the currently more important objects, and less for the context. (e.g. fish-eye views, etc.)
- Dimensional methods. Users can move a focus over a visualization to display different data about the same objects. These methods don't display more objects, but they allow more or different data dimensions of the already displayed ones. (e.g magic lenses, etc.)
- Cue methods. In an existing visualization, objects that meet certain criteria are stressed by assigning visual cues to them so that they are more prominent to the viewer without hiding the context. An example of such a method is to use color saturation and brightness.

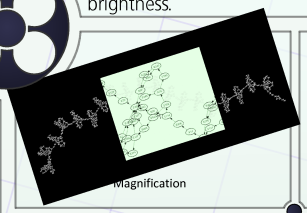
Objectives

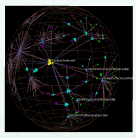
The ultimate goal is to apply current visualization techniques for the information obtained from the simulation. From our primary analysis of this information, we have decided to focus our work on utilizing techniques referred to as "context and focus" methods. The amount of data and information available to visualize in this project is substantial, but what is more important than that is to present this vast amount of inter-related information to showing detail and context simultaneously.

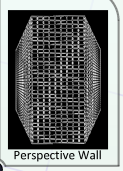
Some this information is as below.

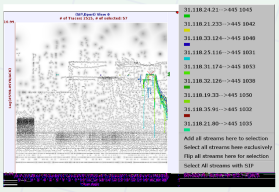
Network topology, Connectivity, Infection dispersion, Affected Hosts Reachability , Network impact, Attacker attributes (IP, DNS resolution) Access paths, Offence presentation, Network traffic statistics Potentially exposed data (IP, Asset, service properties, significance) Specific Host info (no. of vulnerabilities, online time, open ports, etc.) Flow statisticsProtocol statisticsVulnerability infoFirewalls (open ports, blocked IPs, vulnerabilities, exposure, etc.), Security analysis

Examples of Focus and Context


Magnification


3D Hyperbolic Space


Perspective Wall


Histographs

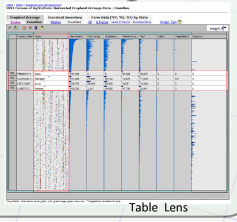


Table Lens

network security laboratory

UNIVERSITY OF NEW BRUNSWICK

UNB

Abstracts of 2008 research publications

Fixed-Parameter Tractability of Anonymizing Data by Suppressing Entries

R. Chaytor, P. Evans*, and H.T. Wareham
Proceedings of the 2nd Annual International Conference on Combinatorial Optimization and Applications (COCOA 2008), Springer-Verlag LNCS 5165 (2008), 23-31.
Abstract
A popular model for protecting privacy when person-specific data is released is k-anonymity. A dataset is k-anonymous if each record is identical to at least (k ? 1) other records in the dataset. The basic k- anonymization problem, which minimizes the number of dataset entries that must be suppressed to achieve k-anonymity, is NP-hard and hence not solvable both quickly and optimally in general. We apply parameterized complexity analysis to explore algorithmic options for restricted versions of this problem that occur in practice. We present the first fixed-parameter algorithms for this problem and identify key techniques that can be applied to this and other k-anonymization problems.

Using Behavioral Specification for Digital System Design

Ke Deng, Eric E. Aubanel, and Kenneth B. Kent
TR08-189, University of New Brunswick, 65 pages, January 2008.
Abstract
This report documents experiences on using behavioral specification for digital system design from the viewpoint of a computer science student with limited knowledge in hardware. The first three sections of this report review the background and basics of OpenMP, VHDL and Handel-C individually. Each of these three sections includes discussion of a related implementation example, which examines practical considerations. In addition, the OpenMP section includes background on parallel computing and its specifications; the VHDL section also discusses fundamental concepts of digital system design; the Handel-C section also includes background on Field Programmable Gate Arrays (FPGAs). The last section compares OpenMP and Handel-C with VHDL and comments on the results.

A Resource Discovery Framework for Semantic Grids Based on the Interface-Based Modular Ontology Formalism

F. Ensan, and Weichang Du
SKG '08. Fourth International Conference on Semantics, Knowledge and Grid, 2008.
Abstract
Semantic grids refer to those grids that their resources and services have been described by the means of semantic Meta data and ontologies. In this paper we propose a resource discovery framework for semantic grids using the notion of modular ontologies. We exploit interface-based modular ontology formalism whose through ontologies can be described and be accessed by a set of interfaces. We show how this formalism help looking for distributed resources in semantic grids. We describe the architecture of the resource manager's nodes and their resource discovery algorithms.

An Architecture and Formalism for Handling Modular Ontologies

Faezeh Ensan
, Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (2008)
Abstract
The goal of my ongoing work is to provide an architecture for developing and manipulating modular ontologies in such a way that each ontology module can plug into or unplug from an ontology. This architecture builds on top of a fundamental formalism for modular ontologies. Through this formalism we are able to define mechanisms for integrating different modules and develop algorithms for reasoning over the integrated modules. The resolution of inconsistencies arisen by conflicting axioms in different modules as well as the investigation of the impact of changes in a module on the other ontology modules are two important issues that need to be taken into consideration during the development of the formalism. Here, we briefly review the overall structure of the research work that I intended to conduct.

Aspects of Inconsistency Resolution in Modular Ontologies

Faezeh Ensan, and Weichang Du
Advances in Artificial Intelligence, 21st Conference of the Canadian Society for Computational Studies of Intelligence, Canadian AI 2008 Windsor, Canada, May 28-30, 2008 Proceedings.
Abstract
Modularization entails more efficient reasoning and better performance in the ontology manipulation process. Therefore, the development of modular ontologies has recently received much attention. One of the most important issues in modular ontologies is dealing with inconsistencies. An inconsistent module may affect the other modules and cause a modular ontology to become inconsistent. Furthermore, the integration of different consistent modules may also result in inconsistency. In this paper, we investigate various types of inconsistencies in modular ontologies. We mostly focus on an interface-based ontology modularity formalism and propose a strategy and an algorithm for isolating inconsistent modules and resolving inconsistencies arisen from the integration of different ontology modules.

Formalizing Ontology Modularization through the Notion of Interfaces

Faezeh Ensan
Knowledge Engineering: Practice and Patterns, 16th International Conference, EKAW 2008, Acitrezza, Italy, September 29 - October 2, 2008.
Abstract
In this paper, we propose a new formalism for modular ontologies, which exploits the notion of interfaces as well as epistemic queries. In the proposed formalism, each ontology module both employs and realizes two distinct sets of interfaces. The axioms in each interface form the public section of the ontology module, while its ABox and TBoxes are private and can only be accessed through epistemic queries. This formalism permits the separation of configuration and development time manipulation tasks of a modular ontology development process. Hence, ontology modules can be developed independently of each others' signature and description language.

Formalizing the Role of Goals in the Development of Domain-Specific Ontological Frameworks

Faezeh Ensan, and Weichang Du
Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008), 2008
Abstract
In this paper we propose a high-level scheme that assists ontology engineers develop appropriate ontological frame- works. By ontological frameworks we mean those structures that specify particular phases and also provide implemented components for developing ontologies. Based on the i* conceptual modeling framework, our proposed scheme guides ontology engineers by customizing a suitable onto- logical framework based on their preferences and their speficific domain necessities. In the proposed scheme, We specify the users of an ontological framework, their high-level softgoals as well as the goals that contribute to these softgoals. We exploit business processes and bind them to the goals in order to implement the framework.

An Interface-Based Ontology Modularization Framework for Knowledge Encapsulation

Faezeh Ensan, and Weichang Du
7th International Semantic Web Conference, ISWC 2008, Karlsruhe, Germany, October 26-30, 2008.
Abstract
In this paper, we present a framework for developing ontologies in a modular manner, which is based on the notions of interfaces and knowledge encapsulation. Within the context of this framework, an ontology can be defined and developed as a set of ontology modules that can access the knowledge bases of the others through their well-defined interfaces. An important implication of the proposed

framework is that ontology modules can be developed completely independent of each others' signature and language. Such modules are free to only utilize the required knowledge segments of the others. We describe the interface-based modular ontology formalism, which theoretically supports this framework and present its distinctive features compared to the exiting modular ontology formalisms. We also describe the real-world design and implementation of the framework for creating modular ontologies by extending OWL-DL and modifying the Swoop interfaces and reasoners.

Agility DK Tutorial with the Amirix AP1100

Farnaz Gharibian, and Kenneth B. Kent
ICI-201, ver. 1.0, Canadian Microelectronics Corporation, 85 pages, September 24, 2008.
Abstract
No abstract.

An Embedded Decryption/Decompression Engine using Handel-C

Farnaz Gharibian, and Kenneth B. Kent
2008 IEEE International Symposium on Industrial Embedded Systems, Montpellier, France, pp. 51-57, June 11-13, 2008.
Abstract
Speed and security of data streams are two key factors in different areas such as data communication and multimedia. Compression algorithms are applied to data streams to increase their communication speed while encryption algorithms are used for assuring the security of the data transfer. AES and LZ77 are two well known algorithms for data encryption and compression respectively. In this paper we propose a model to implement both algorithms, decryption and decompression, in a Field Programmable Gate Array chip. Such a design must address the issues of optimal resource usage of the FPGA, and balance between the throughput of both algorithms. Handel-C is considered as the specification language for this design.

Agility DK Tutorial with the Amirix AP1100

Farnaz Gharibian, and Kenneth B. Kent
ICI-201, ver. 1.0, Canadian Microelectronics Corporation, 85 pages, September 24, 2008.
Abstract
No abstract.

Embedded Systems: New Challenges and Future Directions

Fabiano Hessel , Kenneth B. Kent and Dionisis Pnevimatikatos
ACM Transactions on Embedded Computing Systems, vol. 7, issue 4, article 37, pp. 1-3, July 2008.

Abstract
No abstract.

Application Specific Instruction Sets and their Impact on the Design Space

Kenneth B. Kent, Joseph C. Libby, and Ryan Wood
2008 IEEE Rapid Systems Prototyping Symposium, Monterey, USA, pp. 175-181, June 2-5, 2008.
Abstract
The widespread availability of Field Programmable Gate Arrays (FPGA) coupled with different implementations of "soft-core" processors has created a need to find new methods for optimizing these processors. Because design space is limited on most FPGA's and the maximum clock rate of these processors is heavily bound to the overall size and resource usage it is necessary to find ways to minimize the size of the processor. One such way to minimize the size of a "soft-core" processor is to customize the instruction set on which it operates. Removing instructions that are supported but not utilized by target applications may provide a reduction in design space usage as well as an increase in maximum clock frequencies for the processor.

Determining the Optimal FPGA Design for Computing Highly Parallel Problems

Kenneth B. Kent, and Jacqueline E. Rice
to appear in IET Computer and Digital Techniques journal (15 pages), September 2008.
Abstract
Recongurable hardware has recently shown itself to be an appropriate solution to speeding up problems that are highly dependent on a particular complex or repetitive sub-algorithm. In most cases these types of solutions lend themselves well to parallel solutions. We investigate the optimal design, maximizing performance while existing within the target FPGA resources, on FPGAs for problems with algorithms or sub-algorithms that can be highly parallelized.

Automatic Identification of Parallelism in Handel-C

Joseph C. Libby, Farnaz Gharibian, and Kenneth B. Kent
2008 Euromicro Digital System Design Symposium , Parma, Italy, pp. 660-664, September 3-5, 2008.
Abstract
High level hardware design languages are making it possible for people with little background in hardware design to create their own custom hardware. This allows software designers to begin looking beyond general purpose computing into the realm of customized hardware in order to increase the performance of their applications. The ease with which hardware can be developed using hardware definition languages comes with a cost. Developers accustomed to working in software environments may have issues dealing with some of the more complex facets of hardware

design, such as exploiting parallelism. This work aims to alleviate some of the frustration that may occur when attempting to identify and exploit parallelism in a hardware design by providing a set of tools that can automatically identify parallelism in Handel-C hardware designs.

An Embedded Implementation of the Common Language Infrastructure

Joseph C. Libby, and Kenneth B. Kent
to appear in Elsevier Journal of System Architectures (13 pages), September 2008.
Abstract
The Common Language Infrastructure (CLI) provides a unified instruction set which may be targeted by a variety of high level language compilers. This unified instruction set simplifies the construction of compilers and gives application designers the ability to choose the high level programming language that best suits the problem being solved. While the Common Language Infrastructure solves many problems related to design of applications and compilers, it is not without its own problems. The Common Language Infrastructure is based upon a virtual machine, much like the Java Virtual Machine. This requires that all instructions being executed on the Common Language Infrastructure be translated to native machine instructions before they can be executed on the host processor. This leads to degradation in performance. In order to overcome this problem it is proposed that an embedded processor capable of natively executing the CLI instruction set be developed. The objective of this work is the design and implementation, using VHDL and simulation, of an embedded processor capable of natively executing the CLI instruction set. This processor provides a platform easily targeted by software developers.

Automated Extraction of Concurrency and Pipelined Data Paths in Handel-C

Joseph C. Libby, and Kenneth B. Kent
Design Automation Conference (DAC) High-Level Synthesis: Back To The Future Workshop 2008, Anaheim, USA, 1 page, June 8, 2008.
Abstract
No abstract.

A Handel-C Implementation of a Computationally Intensive Problem in GF(3)

Jonathan Lutes, Joseph C. Libby, and Kenneth B. Kent
International Conference on Advances in Electronics and Micro-electronics, Valencia, Spain, pp. 36-41, September 29 - October 4, 2008. - **Received Best Paper Award**.
Abstract
Computing the irreducible and primitive polynomials under GF(3) is a computationally intensive

task. A hardware implementation of this algorithm should prove to increase performance, reducing the time needed to perform the computation. Previous work explored the viability of a co-designed approach to this problem and this work continues addressing the problem by moving the entire algorithm into hardware. Handel-C was chosen as the hardware description language for this work due to its similarities with ANSI C used in the software implementation.

Service Composition for GIS

Sai Ma, Minruo Li, and Weichang Du
pp.168-175, 2008 IEEE Congress on Services - Part I, 2008.
Abstract
A Geographical Information System (GIS) is a system that captures, analyzes, and manages any spatially referenced data. One common problem in the GIS community is how to generate and publish customized web maps. The existing solutions either deal with spatial data directly which does not allow for applying the customized features, or require and rely on advanced and specialized programming skills. We believe that applying Service Oriented Architecture (SOA) to GIS can improve the interoperability of different GISs and can combine different GISs to provide customized web maps using a web service orchestration language. In this paper, we present a novel solution that applies SOA and Business Process Execution Language (BPEL) to orchestrate web map services into a customized web map. The process of requesting a map layer from a map service provider is an invocation of the remote GIS map service. The process of generating a customized web map becomes a process of combining different GIS map services into a BEPL process. This makes it possible to generate the business logic in BPEL first and then execute it to obtain a new map. Ideally, once the process is generated in BPEL, it can be plugged into any GIS system. This new solution generates a single new map after all layers are combined together, while the existing Asynchronous JavaScript and XML (AJAX) based solution gives a stack of map layers and the layers cannot be saved as one map. We have implemented a framework for the map creator to combine map layers published by different map service providers into a single new map, save the map composition process logic, and publish the new map as a service. Also, the framework provides map brokers more control of and easier interaction with the map composition process.

Flexible Software-Hardware Network Intrusion Detection System

Chen Nan, Ryan Proudfoot, Eric E. Aubanel, and Kenneth B. Kent
2008 IEEE Rapid Systems Prototyping Symposium, Monterey, USA, pp. 182-188, June 2-5, 2008.
Abstract
Network Intrusion Detection Systems (NIDS) and Quality of Service (QoS) demands have been steadily increasing over the past few years. Current solutions using software become inefficient running on high speed high volume networks and will end up dropping packets. Hardware solutions are available and result in much higher efficiency but present problems such as flexibility and cost. Our proposed system uses a modified version of Snort, a robust widely deployed open sourced NIDS. It has been found that Snort spends at least 30% - 60% of its processing time doing pattern matching. Our proposed system runs Snort in software until it gets to the pattern matching function

and then offloads that processing to the Field Programmable Gate Array (FPGA). The software can then go on to other processing while it waits for the results from the FPGA. The hardware is able to process data at up to 1.7GB/s on one Xilinx XC2VP100 FPGA. Our system is more flexible than other FPGA string matching designs in that the rules are not hardcoded. The design is scaleable and will allow for multiple FPGAs to be used in parallel to increase the processing speed even further.

Predicting User Preferences via Similarity-Based Clustering

Mian Qin, Scott Buffett and Michael W. Fleming
Proceedings of the 21st Canadian Conference on Artificial Intelligence, pages 222-233(2008).
Abstract
This paper explores the idea of clustering partial preference relations as a means for agent prediction of users' preferences. Due to the high number of possible outcomes in a typical scenario, such as an automated negotiation session, elicitation techniques can provide only a sparse specification of a user's preferences. By clustering similar users together, we exploit the notion that people with common preferences over a given set of outcomes will likely have common interests over other outcomes. New preferences for a user can thus be predicted with a high degree of confidence by examining preferences of other users in the same cluster. Experiments on the MovieLens dataset show that preferences can be predicted independently with 70-80% accuracy. We also show how an error-correcting procedure can boost accuracy to as high as 98%.

Identifying Sources of Intractability in Cognitive Models: An Illustration using Analogical Structure Mapping

I. van Rooij*, P. Evans, M. Muller, J. Gedge, and H.T. Wareham
Proceedings of the 30th Annual Conference of the Cognitive Science Society (CogSci 2008), 915-920.
Abstract
Many computational models in cognitive science and arti?cial intelligence face the problem of computational intractability when assumed to operate for unrestricted input domains. Tractability may be achieved by restricting the input domain, but some degree of generality is typically required to model human-like intelligence. Moreover, it is often non-obvious which restrictions will render a model tractable or not. We present an analytical tool that can be used to identify sources of intractability in a model's input domain. For our illustration, we use Gentner's Structure-Mapping Theory of analogy as a running example.

A Novel Covariance Matrix Based Approach for Detecting Network Anomalies

Mahbod Tavallaee, Wei Lu, Shah Arif Iqbal, and Ali A. Ghorbani
Sixth Annual Conference on Communication Networks and Services Research (CNSR'08), pages

75-81
Abstract
During the last decade, anomaly detection has attracted the attention of many researchers to overcome the weakness of signature-based IDSs in detecting novel attacks. However, having a relatively high false alarm rate, anomaly detection has not been wildly used in real networks. In this paper, we have proposed a novel anomaly detection scheme using the correlation information contained in groups of network traffic samples. Our experimental results show promising detection rates while maintaining false positives at very low rates.

Detecting Network Anomalies Using Different Wavelet Basis Functions

Wei Lu, Mahbod Tavallaee, and Ali A. Ghorbani
Sixth Annual Conference on Communication Networks and Services Research (CNSR'08), pages 149-156
Abstract
Signal processing techniques have been applied recently for analyzing and detecting network anomalies due to their potential to find novel or unknown intrusions. In this paper, we present a novel network anomaly detection approach based on wavelet analysis, approximate autoregressive and outlier detection techniques. In order to characterize network traffic behaviors, we proposed fifteen features and applied them as the input signals in our wavelet-based approach. We then evaluate our approach with the 1999 DARPA intrusion detection dataset and conduct a comprehensive comparison for four different typical wavelet basis functions on detecting network intrusions. Our work aims to unveil a question when applying wavelet techniques for detecting network attacks, that is "do wavelet basis functions have an important impact on the intrusion detection performance?". Moreover, to the best of our knowledge, the work is the first to analyze the 1999 DARPA's network traffic using flow data instead of its original raw packet data.

Criterion for Intensification and Diversification in Local Search for SAT

W. Wei, C. M. Li, and H. Zhang
Journal on Satisfiability, Boolean Modeling and Computation (JSAT), special issue on SAT 2007 competitions and evaluations, June 2008, volume 4, pages 219-237. ISSN1574-0617.
Abstract
We propose a new switching criterion, namely the evenness or unevenness of the distribution of variable weights, and use this criterion to combine intensification and diversification in local search for SAT. We refer to the ways in which state-of-the-art local search algorithms adaptG2WSATP and VW select a variable to flip, as heuristic adaptG2WSATP and heuristic VW, respectively. To evaluate the effectiveness of this criterion, we apply it to heuristic adaptG2WSATP and heuristic VW, in which the former intensifies the search better than the latter, and the latter diversifies the search better than the former. The resulting local search algorithm, which switches between

heuristic adaptG2WSATP and heuristic VW in every step according to this criterion, is called Hybrid. Our experimental results show that, on a broad range of SAT instances presented in this paper, Hybrid inherits the strengths of adaptG2WSATP and VW, and exhibits generally better performance than adaptG2WSATP and VW. In addition, Hybrid compares favorably with state-of-the-art local search algorithm R+adaptNovelty on these instances. Furthermore, without any manual tuning parameters, Hybrid solves each of these instances in a reasonable time, while adaptG2WSATP, VW, and R+adaptNovelty have difficulty on some of these instances.

Switching Among Non-Weighting, Clause Weighting, and Variable Weighting in Local Search for SAT

W. Wei, C. M. Li, and H. Zhang
In Proceedings of the 14th International Conference on Principles and Practice of Constraint Programming (CP 2008), pages 313-326. Springer. LNCS 5202. September 14-18, Sydney, Australia.
Abstract
One way to design a local search algorithm that is effective on many types of instances is allowing this algorithm to switch among heuristics. In this paper, we refer to the way in which non-weighting algorithm adaptG2WSAT+ selects a variable to flip, as heuristic adaptG2WSAT+, the way in which clause weighting algorithm RSAPS selects a variable to flip, as heuristic RSAPS, and the way in which variable weighting algorithm VW selects a variable to flip, as heuristic VW. We propose a new switching criterion: the evenness or unevenness of the distribution of clause weights. We apply this criterion, along with another switching criterion previously proposed, to heuristic adaptG2WSAT+, heuristic RSAPS, and heuristic VW. The resulting local search algorithm, which adaptively switches among these three heuristics in every search step according to these two criteria to intensify or diversify the search when necessary, is called NCVW (Non-, Clause, and Variable Weighting). Experimental results show that NCVW is generally effective on a wide range of instances while adaptG2WSAT+, RSAPS, VW, and gNovelty+ and adaptG2WSAT0, which won the gold and silver medals, respectively, in the satisfiable random category in the SAT 2007 competition are not.

Uncertainty Treatment in the Rule Interchange Format: From Encoding to Extension

Zhao, J. and Boley, H.
Proceedings of the ISWC 2008 Workshop on Uncertainty Reasoning for the Semantic Web
Abstract
The Rule Interchange Format (RIF) is an emerging W3C format that allows rules to be exchanged between rule systems. Uncertainty is an intrinsic feature of real world knowledge, hence it is important to take it into account when building logic rule formalisms. However, the set of truth values in the Basic Logic Dialect (RIF-BLD) currently consists of only two values (t and f). In this paper, we first present two techniques of encoding uncertain knowledge and its fuzzy semantics in RIF-BLD presentation syntax. We then propose an extension leading to an Uncertainty Rule

Dialect (RIF-URD) to support a direct representation of uncertain knowledge. In addition, rules in Logic Programs (LP) are often used in combination with the other widely-used knowledge representation formalism of the Semantic Web, namely Description Logics (DL), in order to provide greater expressive power. To prepare DL as well as LP extensions, we present a fuzzy extension to Description Logic Programs (DLP), called Fuzzy DLP, and discuss its mapping to RIF. Such a formalism not only combines DL with LP, as in DLP, but also supports uncertain knowledge representation.

Combining Fuzzy Description Logics and Fuzzy Logic Programs

Zhao, J. and Boley, H.
Proceedings of IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology(WI-IAT 2008)
Abstract
Integrating rules and ontologies has become a key requirement for applications in the Semantic Web. Web applications in general have also motivated another requirement, that of handling uncertainty, an intrinsic feature of the real world. In this paper, we present a fuzzy extension to Description Logic Programs (DLP), called fhDLP. fhDLP not only combines DL with LP, as in DLP, but also supports uncertainty representation. More specifically, fuzzy hybrid knowledge bases layered on fhDLP consist of fuzzy hybrid rules with embedded DL queries to fuzzy DL concepts, roles, and axioms.

Abstracts of 2008 PhD Theses

The Collaborative Development of Para-consistent Conceptual Models Influenced by Uncertainty: A Belief-theoretic Approach

By Ebrahim Bagheri
Supervisor: Ali A. Ghorbani
Abstract
The high complexity and diversity of today's design projects demands the participation of multiple experts. The participating experts can influence the design process by sharing their perspective, expertise and resources. The involvement of various experts is often known as collaborative modeling and design. A collaborative modeling environment can encompass various geographical or organizational boundaries. Such collaboration between experts can result in outcomes that may be in practice either inconsistent, vague or incomplete. In this thesis, we provide a correspondence between software conceptual models and annotated propositional belief bases. Through this analogy, we are able to analyze the contents of a given set of software conceptual models, which have been developed by the participants of a collaborative modeling process, known as viewpoints, and specify whether they are incomplete, incoherent, or inconsistent under a closed-world reasoning assumption. Based on the software conceptual models' properties introduced in this thesis, we define an integration game through which the possible inconsistencies of the software conceptual models are resolved. The game consists of several rounds of negotiation and is performed by two main functions, namely choice and enhancement functions. The outcome of this game is a set of inconsistency-free software conceptual models that can be easily integrated to form a unique fair representative of the opinions of the participants. The contributions of the work in this thesis can be briefly enumerated as follows: 1) the development of Subjective belief bases that address uncertainty; 2) the formalization of a multi-stage belief integration game for the integration of multiple Subjective belief bases; and 3) the provisioning of an analogy between Subjective belief bases and software conceptual models. The proposed models are implemented in two Eclipse plug-ins and are thoroughly evaluated from various perspectives. We evaluate our proposed processes through a real-world case study where a group of Computer Science graduate students participate in the study. In addition, we employ a multi-agent simulation to test the convergence and effectiveness of the introduced formal concepts of the thesis. The evaluations have been performed on the basis of the accuracy, precision, and recall of the final conceptual models, as well as the four scales of the Computer System Usability Questionnaire. The results of the evaluations have been reported in this thesis.

A Framework for User Guidance in Web Search Engine Interfaces Based on Past Users Behavior

By Mohammadreza Barouni-Ebrahimi
Supervisor: Ali A. Ghorbani
Abstract
In this thesis, an adaptive Web search engine model is developed that assists its users in preparing relevant queries by recommending the related frequent phrases mined from previous submitted queries. The model also reorders the recommended pages of the conventional Web search engines based on the users interests. Search engine query log mining has evolved overtime to data stream mining due to the endless and continuous sequence of queries received by the search engines known as query stream. We propose an Online Frequent Sequence Discovery (OFSD) algorithm to extract frequent phrases from within query streams based on a new frequency rate metric which is suitable for query stream mining. OFSD is an online, single pass and real-time frequent sequence miner appropriate for data streams. The frequent phrases extracted by the OFSD algorithm are used to guide novice users complete their search queries more efficiently. A re-rank method for the retrieved pages of a conventionalWeb search engine is also proposed which relies on past users clicks for each frequent phrase extracted by OFSD. The contribution of our proposed model is three-fold. First, a Complementary Phrase Recommender module suggests a list of complementary phrases that are syntactically compatible with the entered query segment. Second, a Semantic Phrase Adviser module provides a list of the phrases that are semantically related to the entered query segment. These two modules help the user enter the most related phrases to his/her intention as a query. Third, a Page Rank Reviser module refines the order of the recommended documents prepared by a conventional Web search engine to help the user find the related Web pages on top of the list. Two query logs with different characteristics are used to evaluate the proposed model. The experimental results confirm the significant benefit of monitoring frequent phrases within the queries instead of using the whole query as a non-separable item. The number of the monitored elements substantially decreases, which results in smaller memory consumption as well as better performance. YourEye, our implemented adaptive Web search engine based on the proposed model adjusted for the University of New Brunswick is introduced. Evaluation of YourEye by real users confirms the efficiency of the proposed model in performance as well as user satisfaction.

Multidimensional Programs on Distributed Parallel Computers: Analysis and Implementation

By Khaled M. Ben Hamed
Supervisor:Weichang Du
Abstract
This thesis presents analysis and efficient implementation of programs in multidimensional programming languages on distributed parallel computers. By applying program analysis results, we design a distributed scheduling algorithm for efficient execution of parallel tasks. We perform analysis of multidimensional programs (MPs) to study and collect information on parallelism and dependence, especially context parallelism and dependence. We define and implement an abstract distributed

eduction machine (ADEM) on which multidimensional programs run. From the analysis perspective, a multidimensional program is evaluated in an implicit context space in which its computation values vary. Our first goal of program analysis is studying and collecting information on context parallelism and dependence present among expressions in a multidimensional program. Then, we use the result of the multidimensional program analysis to develop new scheduling strategies for parallel tasks on an ADEM. In this research, we used an abstract dependence graph (ADG) since using a dependence graph where a vertex represents a variable at a concrete context is impractical due to space constraints. This ADG is used by the analyzer to capture and collect information on parallelism and dependence to efficiently schedule parallel tasks onto the ADEM using a heuristic scheduling algorithm. From the implementation perspective, using the result of the program analysis, we investigate a prototype analyzer that takes a multidimensional program and produces program dependence and other scheduling information. This information is used to identify parallel tasks in MP programs and determine how parallel tasks can be efficiently scheduled on the ADEM. The result is a static and/or dynamic scheduling strategy. The experiments we conducted show that the scheduling strategies substantially improve the performance when compared with the performance of other parallel implementations of multidimensional programs.

A Fuzzy Feature Evaluation Framework for Network Intrusion Detection

By Iosif-Viorel Onut
Supervisor: Ali Ghorbani
Abstract
The design of a Network Intrusion Detection System (NIDS) is a delicate process which requires the successful completion of numerous design stages. The feature selection stage is one of the first steps that needs to be addressed, and can be considered among the top most important ones. If this step is not carefully considered the overall performance of the NIDS will greatly suffer, regardless of the detection technique, or any other algorithms that the NIDS is using. The most common approach for selecting the network features is to use expert knowledge to reason about the selection process. However, this approach is not deterministic, thus, in most cases researchers end-up with completely different set of important features for the detection process. Furthermore, the lack of a generally accepted feature classification schema forces different researchers to use different names for the same (subsets of) features, or the same name for completely different ones. It is our believe that these issues are not sufficiently studied and explored by the network security research community. This thesis focuses on mining the most useful network features for attack detection. Accordingly, we propose a new network feature classification schema as well as a mathematical feature evaluation procedure that help us identify the most useful features that can be extracted from network packets.The network feature classification schema is intended to provide a better understanding, and enforce a new standard, upon the features that can be extracted from network packets, and their relationships. The classification has a set 27 classes of features based on the network abstractions that they refer to (e.g., host,network, connection, etc). We use our feature classification schema to select a comprehensive set of 671 features for conducting and reporting our experimental findings. The feature evaluation procedure provide a

deterministic approach for pinpointing those network features that are indeed useful in the attack detection process. The procedure uses mathematical, statistical and fuzzy logic techniques to rank the participation of individual features into the detection process. In particular, we propose a new feature dependency measure for independent evaluation criteria that is, to our knowledge, a pioneer method designed for intrusion detection. In our research we have identified several tuning parameters that directly influence the detection performance of each individual feature. To address this issue, our method takes into account the performance of each feature while using multiple tunings, making the evaluation process more robust to biases that could be accidentally introduced by a poor tuning combination. The experimental results, conducted on three different real-world network datasets, empirically confirm that our feature evaluation model can successfully be applied to mine the importance of a feature in the detection process.

Abstracts of 2008 MCS Theses

Generating Secure Elliptic Curves Over Binary Fields

By Peter Anderson
Supervisor: Rodney H. Cooper
Abstract
Elliptic Curve Cryptography is a form of public key cryptography that offers good security and smaller key sizes then competing methods. The foundation of a secure elliptic curve cryptography system is a well-chosen curve. There are many considerations when generating a curve, the main one being its order. This is often the most costly part of an implementation of an elliptic curve cryptography system and because of this much research has been completed in this area. This thesis has two main goals. First, to offer a solid background in the mathematics of finite fields and elliptic curves so that the reader will understand the advanced topics. Second, to explain and demonstrate the requirements for choosing a secure elliptic curve, with an emphasis on finding the order of the curve and avoiding known attacks.

A Combined Approach for Search of Learning Objects on the Web

By Hamidreza Baghi
Supervisor: Yevgen Biletskiy and Michael Fleming
Abstract
The present thesis describes a system for search and delivery of learning objects. This system combines different methods of search: keyword- and concept based search and personalization. The keyword- and concept-based search methods determine the relevance of each learning object to the query. The personalized search of learning objects determines relevance of each document based on a comparison of a learner (user) profile and learning object descriptions. Such a comparison is based not only on the values of characteristics of the learner profile and attributes of the learning object descriptions, but also the importance of these characteristics and attributes for the learner. The relevance of learning objects to the learner query is determined using a combination of these relevance measures. The approach is evaluated in order to demonstrate its effectiveness and find optimal weighting coefficients.

Improving an OpenMP-based Circuit Design Tool

By Tim F. Beatty
Supervisor: Eric Aubanel and Kenneth Kent
Abstract
As transistor density grows, increasingly complex hardware designs may be implemented. In order to manage this complexity, hardware design must be performed at a higher level of abstraction. High level synthesis enables the automatic conversion of algorithms into hardware implementations, abstracting away the underlying complexities of hardware from the designer. A number of high level synthesis tools have recently been developed, including an OpenMP to Handel-C translator. Improvements to the translator, including a new compiler directive allowing customizable register width, are described. A set of benchmark tests show a decrease in circuit size and increase in performance when the new compiler directive is used.

Computational Grid Emulation for Performance Analysis of Mesh Partitioners

By Basile Clout
Supervisor: Eric Aubanel
Abstract
Mesh-based parallel applications, such as those involving numerical solution of partial differential equations, can take advantage of the processing power of a computational grid. Such applications require a partition, a mapping of the mesh onto the available processors, that optimizes the application execution time. Good partitions can be created with heterogeneous mesh partitioners such as PaGridL. However, mesh partitioners minimize a cost function that does not necessarily reflect the real behavior of an application running a partition on a given computational grid. In order to experimentally compare the quality of partitions created by different mesh partitioners, we implemented Vlan, a heterogeneous computational grid emulator. Vlan can modify the virtual topology of a network and degrade the processor and network performance of a cluster in an accurate, reproducible and independent way. We used these emulated computational grids and a mesh-based benchmark application to analyse and compare the performance of PaGridL with other mesh partitioners. The results show that PaGridL produces partitions of better or comparable quality than other widely used mesh partitioners.

Incorporating Guideline Support Within an Online-Questionnaire Design Tool

By Aaron Cooper
Supervisors: Joanna Lumsden and Jane Fritz
Abstract
Current online-questionnaire design tools do not provide adequate guidance to designers with respect to best practices for online questionnaire design. To investigate, and thereby demonstrate,

how such essential support can be provided, we present a comprehensive case study which focuses on a prototype we developed to incorporate Lumsdens online-questionnaire design guidelines into an existing design tool. After systematically identifying a variety of possible support mechanisms by which we could incorporate the guidelines as an integral part of the existing design tool, we developed a software prototype to demonstrate our selected methods (namely, a critic with a selection of secondary support mechanisms) from both a user interface and architectural perspective. We discuss what we did and decisions we had to make en route to achieving our prototype, reflecting on our research in order to help guide/inform others faced with a similar task.

eTourPlan: A Knowledge-Based Tourist Route and Activity Planner

By Tshering Dema
Supervisors: Harold Boley and Przemyslaw Rafal Pochec
Abstract
Tourism is the worlds largest and fastest growing industry. There are many conventional tourism service providers which are competitively trying to provide the best travel plans and recommendations to customers based on their interests. The Semantic Web is a major endeavour to enhance the Web by enriching its content with semantic (meta)data that can be processed by inference-enabled Web applications. eTourism is a prime candidate for such enrichment, since it is an information-based business. As with any such business, providing the required relevant information for the consumer means a better end product. Thus, providing a well-structured and comprehensive Knowledge Base (KB) for consulting will help bolster eTourism business. In this thesis, we have designed and implemented a KB consisting of tourism domain-specific information. Our KB stores facts about Bhutan, which are structured by a light-weight ontology (adapted from the Harmonise eTourism ontology) and used by partonomy rules that encode the geographical partitioning of tourist regions and provide a basis for activity search capabilities. On top of these, the KBs planning rules are applied to deduce recommendations of routes, activities (attractions and events), and accommodations. This thesis also discusses transferring Friend Of A Friend (FOAF) concepts for semantically describing persons or organizations, to tourist-entity profiles. The FOAFlike Harmonise relation relatedTo between tourist entities is used to chain through Bhutan provinces and attraction profiles and is used to provide attraction-centric recommendations. This prototype, eTourPlan, an eTourism planner using Semantic Web techniques has been implemented in RuleML/POSL as part of this thesis. Results of running eTourPlan in the prototype RuleML engine OO jDREW are reported.

Improving Responsiveness of Sensor Webs

By Ke Deng
Supervisor: Bradford G. Nickerson
Abstract
In this thesis, we present a sensor network programming platform using fuzzy logic.A fuzzy con-

troller model is used to dynamically control the rate of observation of environmental variables. Differing rates arise from changing environmental conditions. Inferencing using linguistic rules provides a compact, human friendly way to represent the knowledge base for controlling environmental sensor networks. Our platform is illustrated with a simulation having two rules and three environmental variables. Our implementation and simulated experimental results show that it is feasible to apply this programming model to wireless sensor networks. The power consumption overhead of fuzzy SWL was approximately 15% when compared with Crossbows benchmark without fuzzy SWL. the RAM and ROM usage of fuzzy SWL increased linearly as the number of rules increased. A novel aspect of this research is the addition of time factors in the fuzzy rules. This permits the fuzzy control system to better model and adapt to slowly changing environmental conditions.

An SSE-Component based Model for RNA Structure

By Mark James Dowe
Supervisor: Patricia Evans
Abstract
Abstract could not be copied

Investigating Resource Estimation for A High-Level Language

By Farnaz Gharibian
Supervisor: Kenneth B.Kent
Abstract
Compression and encryption algorithms are widely studied in transferring data over the networks to satisfy the security and the speed of the data communication. However, the overheads of the compression and encryption algorithms on data transformation have negative affects on real-time data communication. We propose DecRO, a decryption/decompression engine that can be fit in one FPGA. AES and LZ77 are used as decryption and decompression algorithms, respectively. The implementation language for DecRO engine is Handel-C which supports the techniques such as parallelism and pipelining. The implementation results show the efficiency of the engine in using small number of resources while achieving real time performance. A resource estimation framework for Handel-C is proposed based on our taxonomy in resource estimation algorithms. The proposed framework helps high level programmers improve their design performances and decrease their design development time. Our framework consists of two modules: global estimation and local estimation. Global Estimation module focuses on the accuracy of the whole design process, while Local Estimation is fast to facilitate optimization process.

Improved Competitive Learning Neural Networks for Network Intrusion and Fraud Detection

By John Zhong Lei
Supervisor: Ali A. Ghorbani

Abstract
Along with the continuing growth of e-Commerce in North America, fraud and network intrusion cost e-Commerce companies an overwhelming lost each year. Fraud detection and network intrusion detection become more and more important to online e-Commerce business. However, data mining techniques in this domain are facing the challenges of large scale and high skewness of the data, missing and delay labels, and the continuing change of patterns. In this research, we develop two new clustering algorithms, the Improved Competitive Learning Network (ICLN) and the Supervised Improved Competitive Learning Network (SICLN), for the applications in the area of fraud detection and network intrusion detection. The ICLN is an unsupervised clustering algorithm applying new rules to the the Standard Competitive Learning Neural Network(SCLN). In the ICLN, network neurons are trained to represent the center of the data by a new reward-punishment update rule. The new update rule overcomes the instability of the SCLN. The SICLN is a supervised clustering algorithm further developed from the ICLN by introducing supervised mechanism. In the SICLN, the new supervised update rule utilizes the data labels to guide the training process to achieve a better clustering result. The SICLN can be applied to both labeled and unlabeled data and is highly tolerant to missing or delay labels. Furthermore, the SICLN is completely independent from the initial number of clusters because it is able to reconstruct itself according to the labels of the cluster members. Experimental comparisons on both academic research data and practical realworld data for fraud detection and network intrusion detection demonstrate that the SICLN achieves high performance and outperforms traditional unsupervised clustering algorithms.

Service Oriented Architecture Implementation of OpenGIS Web Processing Service

By Jingguang Li
Supervisor: Weichang Du
Abstract
Geographic Information Systems (GIS) incorporate graphical features with tabular data in order to promote geographic results for spatial data problems. GIS Web Services are web based GIS implementations that handle spatial data exchanging mechanisms over the World Wide Web. Open Geospatial Consortium GIS (OpenGIS) Web Processing Service (WPS) is a web based service that provides client accesses across a network to functions that operate on spatially referenced data. The conventional OpenGIS WPS interface defines access operations (processes) to provide accesses to other GIS applications and services on the web. GIS applications and services on the web can be architectured as business services using service oriented web based Geographic Information Systems. Restructuring OpenGIS WPS services with Service Oriented Architecture (SOA) can provide a great opportunity to improve the conventional OpenGIS WPS systems with high quality. This thesis investigates an SOA based new approach to developing OpenGIS WPS systems. The thesis work includes a service oriented architecture for designing SOA based OpenGIS WPS systems, an implementation framework to implement the service oriented OpenGIS WPS architecture, and an application case study for a real-world service oriented OpenGIS WPS system based on the design process and implementation framework. The evaluation and analysis of the SOA approach compared with the conventional approaches shows that SOA based OpenGIS WPS systems provide

higher quality of services in many aspects, such as modifiability, and reusability.

Web Based Development Environment for GIS Map Services

By Sai Ma
Supervisor: Weichang Du
Abstract
A Geographical Information System (GIS) is a system that captures, analyzes, and manages spatially referenced data. One common problem in the GIS community is how to generate and publish customized web maps. The existing solutions either deal with spatial data directly which does not allow applying the customized features, or requires and relies on advanced and specialized programming skills. In this research, we use Asynchronous JavaScript and XML (AJAX) computing technology to improve performance on viewing dynamic web maps. We apply Service Oriented Architecture (SOA) to GIS systems to improve their interoperability using web service composition technology to provide composite customized web maps. We implement a development environment that builds the both AJAX and SOA based solutions as deliverable web based software systems.

Quality of Service (Qos) for video tranamission

By Shihyon Park
Supervisor: John M. Dedourek
Abstract
There is growing popularity of real time Internet traffic such as audio and video streams. However, traditionally on the Internet, different types of traffic are not distinguished. When congestion occurs, all traffic suffers the same impairments, e.g. increasing delay, more variable delay, and packet loss. However, different types of traffic have differing sensitivities to these impairments. In order to deal with these issues, QoS schemes have been proposed. The primary goal is to provide a testbed and platform for investigating characteristic of real time Internet traffic on the heterogeneous networks. QoS techniques for transmission of real-time video that use a DiffServ mechanism that includes an efficient bandwidth agent to allocate bandwidth on heterogeneous networks, a Hierarchical Token Bucket (HTB) queuing scheme at the output interface of Linux-based routers, and a policing mechanism at the incoming interface of the edge router (ER) in a Linux-based testbed. The characteristics of real-time video traffic using MPEG2 streaming video and VLC for server/client so that we will have a good idea what bandwidth and burst size is required to stream MPEG video through the QoS Diffserv domain. We set an efficient testbed to investigate how real-time streams behave in the QoS scheme, and to provide a realistic recommendation to manage the available bandwidth for both the QoS provider and clients. Tests were conducted with interview.mpg (interview clip), soccer.mpg (sports clip), and card.mpg (entertainment clip). From this test, a real-time streaming videos require a minimum amount of bandwidth, but other real-time streaming videos require a certain size of burst to properly be transmitted.

On the role of temporal and spatial representations in light of ETS formalism

By Benjamin Reuben Peter-Paul
Supervisors: Lev Goldfarb and Weichang Du
Abstract
The Evolving Transformation System (ETS) is a representational formalism for classification of real world objects and their relationships. In ETS all objects are viewed and represented as processes. ETS object representation, as a purely temporal representation, is a temporal sequence of structured events called a struct. Compared to the conventional mathematical, i.e. spatial, representations, it appears to be a primary form of representation, which can be spatially instantiated. Such spatial instantiations can vary considerably, so that the temporal representation could be considered as a more abstract and compact form of representation. This thesis investigates the connection between temporal representation, central to ETS, and spatial representation, central to conventional representational formalisms. To elucidate this connection, we develop and study a family of 3D instantiations of temporal representations for the ETS class of objects called "Bubble Man". To obtain the temporal representations for study, we develop a framework of ETS data structures and algorithms, designed to simulate the ETS class element generation process (in a top-down manner). Then we use a finite state transducer to simulate the physical instantiation of ETS temporal representations. We conclude that spatial representation, offered by conventional formalisms, is subordinate to temporal representation, such that, the information represented in the former may be systematically reproduced from the latter.

Adjustable Autonomy in an Automated Negotiation Agent

By Atteeka Rashid
Supervisors: Michael Fleming and Scott Buffett
Abstract
For an agent negotiating for a suitable deal on a user.s behalf, uncertainty may arise as to whether the user would find a particular offer acceptable. Given the ability to adjust its autonomy, the agent could hand control over to the user instead of making a decision on its own of whether or not to accept the offer, taking into account the benefits (e.g. making an acceptable deal) and costs (e.g. user may not respond; meanwhile, other opportunities will be lost) that could result. The aim of this thesis has been to develop a framework for an agent to reason about autonomy adjustment, and in particular, to determine an optimal level of autonomy to adopt during negotiations. For a given autonomy level, there is a trade-off between the assurance of making an acceptable deal that can come with consulting the user, and the resulting constraints on the agent.s options in pursuing the best possible deal. The approach taken involves formulating the problem of deciding which offers to ask about and which to accept as a Markov Decision Process (MDP) for each possible autonomy level, and then solving and simulating the MDPs to determine which level provides the best trade-off. Tests on simulated data demonstrate that, with the developed framework, agent performance can be improved in automated negotiation systems.

Managing Software Quality in Educational and Small Business Environments

By Khaled Ali M Slhoub
Supervisors: Dawn MacIsaac and M Crease
Abstract
The purpose of this work was to propose a light-weight, learner-centric small-scale strategy for managing software quality in educational and small business environments. The strategy is made up of a development process, a set of quality metrics and a set of associated standards. The development process is based on Agile practices. The quality metrics were generated via a goal-question-metrics process and include metrics for tracking product and process quality, specifically risk, product satisfaction, prototype suitability, prototype development duration, work week, productivity, efficiency, defect density, and maintainability (with respect to coding standards and documentation standards). Associated standards include benchmarks for each metric, and a set of documentation, coding, and logging procedures. A framework for a tool which supports the metric tracking was also developed. The framework, called the Software Quality Resource Tool (SQRT), was designed and implemented as part of a standard eclipse development environment to accept plug-ins which monitor the metrics proposed in the strategy. To demonstrate this utility, a module for complete automated tracking of prototype suitability was designed and implemented.

An Opportunistic Communication Paradigm for Cyber-Engineering

By Mohsin Sohail
Supervisor: Mihaela Ulieru
Abstract
This work is an integral part of the research on eNetworks as infrastructures for the future Cyber-Physical Ecosystems carried on in the Adaptive Risk Management (ARM) Laboratory at UNB. Within this broader context, the aim of my research is to show practically a proof of concept for a network architecture based on Mobile Code for Weisers vision of ubiquitous computing and opportunistic computing on Wireless Sensor Networks (WSN). The concept of "network architecture" is very abstract; it defies rigorous analysis and thorough simulation, and is best understood through experimentation in a realistic environment. Therefore, I plan to design an integral component of the ARM testbed which deals with the integration of new and evolving pervasive networks, namely the Wireless Sensor Networks (WSN) composed of Motes, complemented with another mobile network composed of cell phones, smart phones and/or PDAs. The need for this integration is imminent given the bottlenecks which pose challenges for the current Internet. Additionally, expectations that the Future Internet will be more than simply a source of knowledge through end-to-end connectivity but also an interface between us and our surrounding physical world, calls for a radical transformation of the current Internet. The proposed paradigm in my thesis will provide an opportunity to enhance both mobile and wireless sensor networks by leveraging on each other through novel applications for home and industrial automation. In addition, the proposed paradigm will serve as a foundation for future investigation of interdependencies among heterogeneous large scale networks.

Security and Asynchronous Javascript and XML (AJAX): Assessing the Vulnerability of a Simple AJAX Deployment to a JAVASCRIPT Hijacking Attack

By Elliot Sullivan
Supervisor: Dawn MacIsaac
Abstract
No abstract.

Multi-level Online Learning

By Biao Wang
Supervisor: Bruce Spencer and Huajie Zhang
Abstract
Online recommendation techniques are widely used to improve users' response rates in interactive online marketing. Although some machine learning schemes, e.g., nave Bayes, have been successfully employed for this purpose, each scheme has its advantages and disadvantages. Online learning schemes provide another practical approach to online recommendation, in which a classifier is learned incrementally from examples, interleaving predicting and training. Online learning comes into play when we have repeated interactions. In each iteration, it accepts a request for prediction of a given example, makes a prediction, and observes the true label of the example, and the model is updated to improve later predictions if the observation disagrees with prediction. Online learning can be useful for interacting with people. For example, in online recommendation, although the users' tastes usually remain a constant for a long period of time, their interests may change frequently. In our setting we want to allow the user's evaluation of an object to change during the interaction as we track that changing opinion. Thus, it is necessary to employ an online learning scheme when we observe their changing ratings on objects. In real-world applications, although binary data is sometimes used to represented quantity, it is too coarse in most situations. Multi-level comes into play when we want to predict the multi-level response from a user, for example, multi-level data is required when we want to predict the degree of appreciation a user may have for an object. Therefore, online learning with multi-level predictions while interacting with users is our goal. In this thesis, after presenting to the readers a detailed survey of online learning, linear classification models, and online recommender systems, we propose a multi-level online learning scheme called MWinnow, which is expected to be not sensitive to the curse of dimensionality and have good behavior in the presence of irrelevant attributes, noise, and even a target function changing in time. It is very cheap to implement and can be efficiently applied to online recommender systems. We perform experiments to systematically evaluate the performance of the MWinnow scheme using nave Bayes as the baseline scheme. The results show that MWinnow is at least competitive with nave Bayes and even significantly outperforms it in some circumstances in terms of prediction quality and real-time performance. The MWinnow scheme is promising for future applications, especially as a recommendation scheme.

A Novel Protocol Suite for the Virtual Home Environment in Heterogeneous Networks

By Xi Yuan
Supervisor: Bernd J. Kurz and John DeDourek
Abstract
As the demand for mobile Internet access increases, the concept of Virtual Home Environment (VHE) was introduced to provide the mobile users with consistent access to the value added services they have subscription to while they roam. While the requirements for providing VHE in homogenous networks have been gradually clarified by recent research, the introduction of the consumer-based model, in which the mobile users are no longer associated with fixed network access service providers with long term contracts, raised the new challenge of providing VHE across heterogeneous networks. As the first step towards providing a complete VHE solution in the heterogeneous network environment, a third-party authentication/authorization protocol called VHE Session Authentication Protocol (VHESAP) has been developed as the foundation of other VHE modules. To enable easy protocol implementation and modification, the General Signaling Protocol Interface (GSPI) was developed, featuring a rich set of highly modularized protocol design tools. Using the GSPI implementation of VHESAP, both of CBM and SBM environments are simulated in the lab to evaluate their influence on the protocol performance.

Dynamic Clustering of Large Scale Data Using Random Sampling

By Reza Zafarani
Supervisor: Ali A. Ghorbani
Abstract
Clustering is the unsupervised partitioning of feature vectors into subsets of similar objects. In this thesis, a new framework for clustering large scale datasets based on random sampling is proposed. The framework addresses well known challenges in clustering such as Dynamism, High Dimensionality, Stability, and Scaling. The core of the proposed framework is based on scaling known clustering algorithms for large scale datasets. Furthermore, this algorithm is also equipped with a novel technique for determination of the optimal number of clusters in datasets. These properties add the capabilities of reducing the effect of high dimensionality and scale in datasets to this algorithm. Various experiment have been conducted to analyze the performance of the framework. These experiments are dedicated to the justification of different decisions taken in the design process of the algorithm as well as determining the optimal values of algorithm's parameters and evaluating the clustering algorithm. The experimental results show that the algorithm is not only capable of determining the optimal number of clusters accurately but is also competitive in predicting the true cluster labels.

Assisting Interoperability between Learning Objects and Learners in an E-Advising Scenario

By Luqian Zhu
Supervisor: Dawn MacIsaac and Yevgen Biletskiy
Abstract
Integrating information from diverse heterogeneous environment is a challenging task. To solve the incompatibility problem between educational materials created within varying cultural contexts is especially important for e-Learning environments. The rapid development of e-Learning technologies provides access to a large amount of online learning materials, often wrapped as XML-based learning objects, which originate from multiple cultural backgrounds. However, since knowledge is represented on a global scale, different contexts and incompatibilities between learning objects often create barriers for users, so that limits their exchange efficiency. In this thesis, we focus on assisting Learning Object interoperability in an electronic academic advising (e-Advising) scenario by using Semantic Web techniques and context mediation approach. The experiments and evaluation conducted in translating student's transcript between different schools show that the knowledge representation model and mediation approach we propose can be successfully applied in e-Advising systems.

Author-index

ISBN 978-1-55131-134-0