RWTH Aachen

Department of Computer Science
Technical Report

A Pointillist Approach for Comparing Honeypots

Fabien Pouget and Thorsten Holz

ISSN 0935–3232 · Aachener Informatik Berichte · AIB-2005-05

RWTH Aachen · Department of Computer Science · March 2005

A Pointillist Approach for Comparing Honeypots

Fabien Pouget[1] and Thorsten Holz*[2]

[1] Institut Eurécom, BP 193, 06904 Sophia-Antipolis Cedex, France
[2] Laboratory for Dependable Distributed Systems, RWTH Aachen University, 52056 Aachen, Germany

Abstract. The concept of electronic decoys (*honeypots*), which are network resources that are deployed to be probed, attacked, and eventually compromised, is used in the area of IT security to learn more about attack patterns and attackers' behavior in real-world networks. Our research focuses on gathering detailed statistics on the threats over a long period of time in order to get a better understanding of their characteristics. In this perspective, we are deploying honeypots of different interaction levels in various locations. At a first glance, these honeypots can be considered as permanent sensors that gather statistical information on a long-term perspective.
Generally speaking, honeypots are often classified by their level of interaction. For instance, it is admitted that a high interaction approach is suited for recording hacker shell commands, while a low interaction approach provides limited information on the attackers' activities. So far, there exists no serious comparison to express the level of information on which both approaches differ. Thanks to the environment that we are deploying, we are able to provide a rigorous comparison between the two approaches, both qualitatively and quantitatively. The proposed analysis leads to an interesting study of malicious activities hidden by the noise of less interesting ones. Furthermore, it shows the complementarities of the two approaches: a high interaction honeypot allows controlling the relevance of low interaction honeypot configurations. Thus, both interaction levels are required to build an efficient network of distributed honeypots.

Keywords: Honeypot Interaction, Attack Monitoring, Correlation, Forensics

1 Introduction

Many solutions exist to observe malicious traffic on the Internet. However, they often consist in monitoring at a very large number of honeypots (unused address spaces) to monitor malicious activities. Several names have been used to describe this technique, such as *network telescopes* [Cai05, MVS01], *blackholes* [SMS, CM], *darknets* [Cym04] and *Internet Motion Sensor* (IMS) [CBM+04]. Some other solutions consist in passive measurement of live networks by centralizing and analyzing firewall logs or IDS alerts [Cen, YBJ04]. Coarse-grained interface counters and more fine-grained flow analysis tools such as NetFlow [Sys] offer another readily available source of information.

So far, nobody has investigated the possibility of using a large number of local and similar sensors deployed all over the Internet. However, we strongly believe that local observations can complement the more global ones listed above. A direct analogy can be made here with weathercast or volcanic eruption prediction. As a consequence, we are deploying many honeypot environments in various locations thanks to motivated partners, as part of the Leurre.com Project. The main

* Work by Thorsten Holz was supported by the Deutsche Forschungsgemeinschaft (DFG) as part of the Graduiertenkolleg "Software for mobile communication systems"

objective consists in gathering statistics and precise information on the attacks that occur in the wild on a long-term perspective. We have initially used high interaction honeypots. Then, because of the incoming and increasing number of participants in addition with the hard constraints imposed by their implementations, we have considered the idea of deploying low interaction honeypots. At this time writing, some environments of different interaction levels are running. We invite the interested reader to have a look at the existing publications for more information on that point [DPD04a, PD04a].

The environmental setup we have developed gives us the opportunity to make a rigorous comparison of the different interaction approaches, both qualitatively and quantitatively. So far, there does not exist other comparison like this. Honeypots have been classified in application categories without concrete justification [Spi02a]. For instance, it is admitted that a high interaction approach is suited for recording hacker shell commands, while a low interaction approach provides limited information on the attackers' activities. This paper intends to show this classification is too restrictive. In regards of our research objectives, both approaches present values.

The contributions of this paper are as follows:

- We show that both approaches provide very similar global statistics based on the information we collect.
- A comparison of data collected by both types of environments leads to an interesting study of malicious activities that are hidden by the noise of less interesting ones.
- This analysis highlights the complementarities of the two approaches: a high interaction honeypot allows controlling the relevance of low interaction honeypot configurations. Thus, both interaction levels are required to build an efficient network of distributed honeypots.

The rest of this paper is structured as follows: Section 2 describes and justifies the setup of a distributed honeypot platform. It also briefly depicts the so-called Leurre.com environment. This environment has been implemented in two different ways corresponding to two distinct interaction levels. The analysis is then built on these two approaches. Section 3 makes a comparison on global statistics obtained by means of these two distinct implementations. Particularly, we show the similarity of the information provided by the two environments. In Section 4 we take a closer look at some activities that could potentially be different between platforms. This in-depth study of both platforms leads to the discovery of strange attack scenarii that require a particular attention. We also show that high interaction honeypots can be used as reference points to optimize the configuration of low interaction ones. These two last Sections bring us to explain the motivations behind the Leurre.com project that we are deploying. The last Section concludes this paper.

2 Environment Setup: two different levels of interaction

2.1 High Interaction Experimental Setup - H_1

We have presented in previous publications [DPD04a, DPD04b] some experiments based on so called "high interaction honeypots". this environment, called in the following H_1, is a virtual network built on top of Vmware (see Figure 1) [Cor]. Three machines are attached to a virtual Ethernet switch [3] supporting ARP spoofing. The VMware commercial product enables us to configure them according to our specific needs. mach0 is a Windows98 workstation, mach1 is a Windows NT Server and mach2 is a Linux Redhat 7.3 server. The three virtual guests are built on non-persistent disks [Cor]: changes are lost when virtual machines are powered off or reset. We perform regular reboots to guarantee that the virtual machines are not compromised, as the objectives consist in gathering statistical data in a long-term perspective. A fourth virtual machine is created to collect data in the virtual network. It is also attached to the virtual switch and tcpdump is used as a packet gatherer [uti]. This machine and the VMware host station are totally invisible from outside. Both mach0 and mach2 run an ftp server; in addition, mach1 also provides a static web server. Logs are collected daily and transferred to a centralized place.

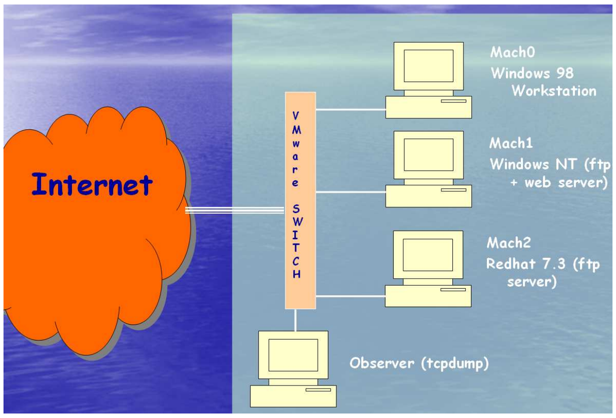


Fig. 1. H_1 Environment scheme

We have also made some comparisons with another "high interaction" honeynet called GenII [The03]. However, the collected data were based on snort-inline alerts. First, alerts provide different information than row data and are quite likely false positives. Second, snort-inline drop packets based on the way it *estimates risk*. These two reasons have prevented us to make interesting comparisons at this stage. We do not refer to this architecture in the following.

2.2 Low Interaction Experimental Setup - H_2

We have deployed a platform called H_2 similar to the one presented before, but with emulated operating systems and services. We have developed it based on some open source utilities. Indeed, it consists in a modified version of Honeyd [hon04]. The platform only needs a single host station, which is carefully secured by means of access controls and integrity checks. Honeyd emulation is limited to a configuration file and a few scripts. It emulates the three same Operating Systems than H_1 for mach0, mach1 and mach2. We have scanned the open ports in H_1 and

[3] A switch in the VMWare jargon actually behaves as a hub

opened the very same ones in the Honeyd configuration file for each three virtual machines. Some service scripts that are available in [hon04] have been linked to open ports, like port 80 (web server) or port 21 (ftp). As a consequence, H_2 interaction refers to a list of open ports and simple scripts. It can be seen as simplistic behavioral model of H_1.
Everyday, we connect to each machine to retrieve traffic logs and check security logs.

2.3 Information Extraction

Dump files are collected from H_1 and H_2. They are stored in a centralized database. They are also analyzed by means of other utilities and this information is collected as well. To make it short, and for a better understanding of the following sections, we list below some information types that can extracted from the dump files:

- IP Geographical location
- Domain name resolution
- Passive OS fingerprinting
- TCP stream analysis
- Etc

We do not want to detail the database architecture here; we invite the interested reader to look at our previous publications, where we have introduced the setup in detail [PDD05].

3 Platforms comparison

3.1 Introduction

To validate models, we need to check their conformance with standard structure behavior. Honeypots can be seen as *black boxes*: they describe a device whose internal structure can be disregarded. All we need to know is that the device transforms given inputs into predictable outputs. It can thus be treated as opaque, as if its contents cannot be seen. If we briefly formalize this: let be I_1 the set of information pieces provided by Honeypot H_1 (the high interaction honeypot). In the same way, let be I_2 the set of information pieces provided by Honeypot H_1 (the low interaction honeypot). Intuitively, we have $I_2 \subset I_1$. However, it is more difficult to estimate on which extent I_2 brings less information. The following Sections intend to qualify and quantify this information difference $I_1 - I_2$.

The initial setting is the following: Honeynets H_1 and H_2 are both placed in the same environment, i.e. the same network, but with different addresses. The virtual machines mach0, mach1 and mach2 have three adjacent IPs in H_1, say X.X.X.1, X.X.X.2, X.X.X.3. In a similar way, virtual machines mach0, mach1 and mach2 in H_2 have contiguous addresses which are very close, resp. X.X.X.6, X.X.X.7, X.X.X.8. Honeynet H_1 has been running since February 2003. Honeynet H_2 started running on July 2004. A technical problem prevented us to collect the whole November 2004 month. Thus, we will focus on data collected on both environments from August 2004 to October 2004, that is 10 continuous weeks.

We propose in the following Section to study the differences between both platforms on that period, thanks to the preprocessed data stored in our database(see Section 2.3).

3.2 Analysis on global statistics

Attack Categories Both environments H_1 and H_2 are targets of attacks. However, each environment contains three virtual machines running different services and different OSs. They are apparently not equally targeted. This leads us to define three major categories of attacks:

- The ones which target only one machine. They are called attacks of Type I.
- The ones which target two out of three virtual machines. They are called attacks of Type II.
- The ones which target the three virtual machines. They are called attacks of Type III

Attack Type	H_1 Environment	H_2 Environment
Total	7150	7364
Type I	4204 (59%)	4544 (62%)
Type II	288 (4%)	278 (4%)
Type III	2658 (37%)	2542 (34%)

Table 1. Different Attack Types observed on H_1 and H_2

Table 3.2 represents the distribution (in percentage) of these 3 categories on each environment H_1 and H_2. Values are very similar. Now, we propose a closer look at Type III attacks. They stem for around 35% of the whole attacks. Figure 3.2 represents the number of associated sources observed on environments H_1 (dark curve)) and H_2 (light curve) every 2 days. Curves have the same general shape. We expected no difference insofar as we have made the assumption in [PD04a] that attacks targeting the three virtual honeypots are to be scans. Thus, scans should be observed independently on the platform. In other words, there should the same number of scans on both platforms. This is not exactly the case on Figure 3.2 where curves have small dissimilarities.

A closer look at the attacks confirms that almost all IP sources associated to Type III attacks have been observed on both environments. For those which are not included in one curve, it appears that they are classified as attacks of type III in one environment, and in attacks of Type II with the other one. In seldom cases, they are even classified as attacks of type I. An analysis of such attacks reveals that they often consist in a single TCP packet sent to one target. It might happen that packets are lost due to congestions in the Internet and we can also imagine that such packets are not retransmitted by the attacker. To validate this assumption, we check that there is any bias in the loss observation, that is, we observe an equal number of packet losses on platform H_1 and on platform H_2. In addition, the number of supposed scan packet losses is distributed among all virtual machines without apparent preferences. We point out that the value

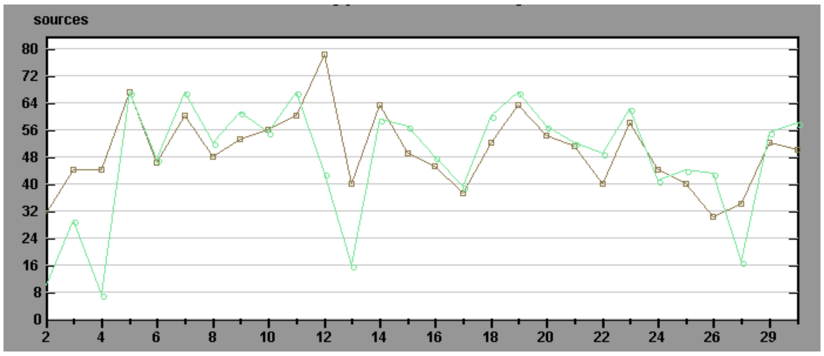


Fig. 2. Attacks of Type III on the two French platforms H_1 and H_2

we observe can be linked to the estimated TCP packet loss value in the path between the attack machine and the honeypot environment at a given date. If for a period of time $\Delta(t)$ the estimated packet loss between the attacking source and the honeypots environment is p_loss, then the probability Pr to get an incomplete scan on the six virtual machines becomes:

$$Pr = 1 - (1 - p\_loss)^6 \tag{1}$$

In our cases, we identify 86 such losses over a total of 2658 type III attacks of the 2-month observation. According to the previous equation, this is equivalent to an average packet loss of 0.6%, which remains coherent with actual traffic monitoring [Cen01]. This is even quite low if we compare with the global average 2-5% observed on the Internet Traffic Report web site [Rep05]. However, we also note on their site high differences between continents. European traffic seems less susceptible, in average, to packet losses than other continents like Asia.

A first assertion is:

Assertion 1 *This is not necessary to deploy honeypots using hundreds of public IP addresses in order to identify scan activities against large block IPs. Three addresses contained in that block are sufficient. Large-scale scans will be attacks on the three honeypot machines. We may observe only two attempts in case of packet losses, as it appears scans do not implement retransmission processes.*

To complete the analysis, we also observe another interesting property based on the fact that virtual machines have being assigned contiguous IP addresses. There is one principal scanning technique which consists in targeting IPs by increasing IP number. To quantify this scanning method, we represent in 3.2 the six possible orders of scanning that have been observed. We give for each of them their frequency (in percentage), that is, the number of Sources which have targeted the three virtual machines over the total number of Sources associated to Type III attacks.

The frequencies remain quite constant ($\simeq 4\%$) over months. Attacks targeting machines by increasing IP number corresponds to 79% of the total. The other frequencies are equally distributed. It is important to point out that all attacks which have targeted one platform in the last 5 orders have scanned the other environment in the increasing sequence order (order 1). Another good indicator

Type III Attack Order	Percentage
Order 1: Mach0, Mach1, Mach2	79%
Order 2: Mach0, Mach2, Mach1	5%
Order 3: Mach1, mach0, Mach2	4%
Order 4: Mach1, Mach2, Mach0	5%
Order 5: Mach2, Mach0, Mach1	3%
Order 6: Mach2, Mach1, Mach0	4%

Table 2. Potential orders of scanning for Type III attacks

to complete this analysis is the observed sequence of ports used by the attacking source on the different virtual machines. It consists in an arithmetic sequence which as a common difference of 1. This simple observation leads to three major remarks:

- We do not observe scan activities that sweep through IP addresses following a sequential decrease.
- All scans that target three consecutive IPs are programmed to hit them in sequential increasing order. It might happen, however, that the order is slightly disrupted because of some packet retransmissions. A simple glance at the attacker source ports confirms this. The non-privileged ports are used sequentially.
- Scanning machines do not wait for a scan to be finished in order to target the next IP. Scanning threads are not blocking. In other words, we observe that temporal periods of scanning activities against two virtual machines from a same source can overlap.

Such information can be interesting for defense preventing systems. If two honeypot sensors are placed on the first/last IPs of a given Network IP range, a preemptive block of all sources having targeted these unused IPs would avoid them to reach the other network machines. It is all the more important that we have shown by means of simple experiments in [PD04a] that global scanning activities are often preliminary steps to accurate attacks on opened ports. We are studying the feasibility of such a system. Finally, we intend to have a closer look at scanner implementation options in order to build relationships with the observed traces. For instance, the advscan Sourceforge Project allows some variables like the number of concurrent threads, the delay or the scanning duration [adv05].

Type II Attack Analysis Attacks of Type II represent a small fraction of all observed attacks. As we explain in the previous Section, some scanning activities that target a large block of IPs can *miss* some addresses insofar as the tools do not retransmit lost packets. It has been observed that 88% of the attacks of type II are residues of scanning attacks on both environments H_1 and H_2, and thus, are incomplete Type III attacks. The remaining 12% are more interesting.

- For 9% of Type II attacks: The IPs have been observed against two virtual machines on one environment, namely mach0 and mach2. The attacking IPs have also been observed on the other environment. A closer look at the attacker source ports leads to the conclusion that these attacks scan one out of two successive IPs. Indeed, all these IPs which have targeted mach0

(X.X.X.1) and mach2 (X.X.X.3) on H_1 have targeted mach1 (X.X.X.7) only on H_2. Inversely, all these IPs which have targeted mach0 (X.X.X.6) and mach2 (X.X.X.8) on H_2 have only targeted mach1 (X.X.X.3) only on H_1. This can be seen as a limitation of our local honeypot platforms.
Indeed, we will not be able to distinguish attacks with larger scan hops. We are not aware of any tool using this strategy. However, a complementary confirmation can be checked by means of large telescopes and blacknets.
- For 3% of Type II attacks: They concern attacks on the sole two Windows machines mach0 and mach1, and on both environments H_1 and H_2. They are for instance attack attempts on port 5554 (Sasser Worm FTP Server [Res04]) or port 9898 (Dabber Worm backdoor [LUR04]). It is clearly not the usual propagation techniques referring to these worms. We face attacks that know a priori the Windows machines on both environments, and that have made some random-like attempts on them. Indeed, we do not observe attempts on both ports but only one on each machine. The attacking IPs are also not observed on both environments, unlike the others.

This leads to a second assertion:

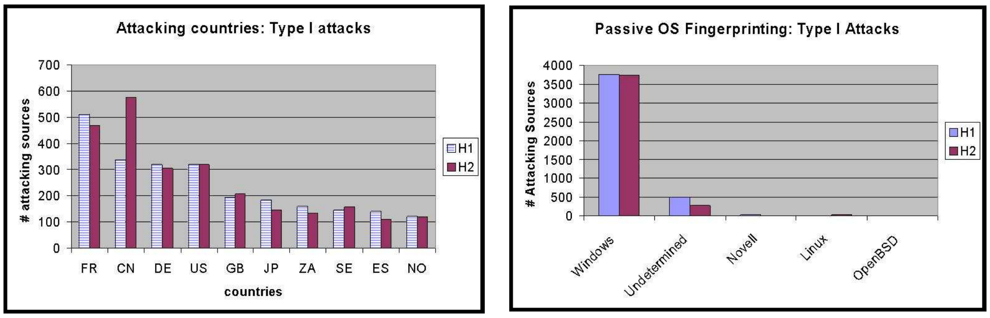
Assertion 2 *Attacks targeting two out of three machines can be specific to the two victim machines, but are with high probability residues of scanning activities.*

Type I Attack Analysis Categories of type I are far more difficult to compare between environments H_1 and H_2. They stem for around 60% of all attacks on both. Figures 3 represent some global characteristics of such attacks on both environments. To be more precise, Figure 3(a) presents the geographical location of the attack sources corresponding to Type I attacks. On the horizontal axis are presented the top 10 countries. The vertical axis gives the number of associated attacking sources for each environment. Figure 3(b) gives the estimated attacking OS, based on passive OS fingerprinting techniques [pof04]. The vertical axis gives also the number of associated attacking sources for each environment.

As a general remark, there is no important differences between environments H_1 and H_2. For instance, both are targeted by 4 main countries with same order of importance (France FR, China CN, Germany DE, United States of America US)[4]. The other country participations are more variable over months but remain coherent between both environments. The passive fingerprinting analysis confirms this similarity between attacks on the two environments too. The IP sources which attack the platforms are essentially running on Windows. To complete this comparison, Figure 3.2 lists the 10 most targeted ports on each platform H_1 and H_2. The vertical axis shows the number of associated attacking sources for each environment. The order is identical and the number of attacks on those 10 ports are very similar on both environments.

In summary, Type I attacks represent lots of common characteristics between platforms H_1 and H_2. On the other hand, the amount of information collected on both environments is totally different. Through the high interaction platform H_1, 480684 packets have been sent against virtual machines. This is 40 times more

[4] The geographical location has been obtained by means of the Maxmind commercial utility [max04]



(a) Attacking Countries (b) Passive Fingerprinting

Fig. 3. Global statistics Comparison between H_1 and H_2

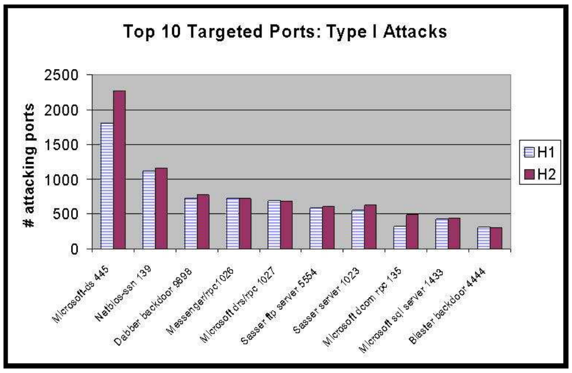


Fig. 4. Top 10 Targeted Ports for Type I attacks on each platform H1 and H2

than through H_2 environment. It is quite normal, insofar as many attacks target *talkative* services like Microsoft ones. The following Section intends to present a refined analysis of the differences which are mainly due Type I attacks.

3.3 Refined Analysis on Type I attack

Different Type I categories As we have shown in the previous Section, Type I attacks present very similar global statistics (see Figures 3 and 3.2). On the other hand, the number of collected packets is totally different. Based on the comparison we make between the two platforms, we thus propose to refine the Type I attack analysis. From the observations that have been made previously, it appears that attacks of Type I can correspond to at least two phenomena. They are listed in the following. The third category gathers all the other *non classified* attacks:

– Sequential Scans residue: This is the first category of Type I attacks. They are to be compared with the same large scanning activities than we presented in Section 3.2. This case can be rare but we can also imagine that two losses can happen on the same environment. It is simply identified by looking at common IP addresses on both environments which have targeted one machine

on one environment and three virtual machines on the other one, during a short period of time. We find the same number of corresponding sources on H_1 and on H_2, 1 out of 1000 Type III attacks in average.
To validate that it correctly corresponds to packet losses, we consider that if for a period $\Delta(t)$ the estimated packet loss between the attacking source and the honeypots environment is *p_loss*, then the probability Pr to observe two losses out of three scans becomes approximatively: $Pr = 3 * p\_loss^2 * (1 - p\_loss)$. This remains coherent with the low number of cases we observe. This category has been observed thanks to the complementarities between H_1 and H_2. Indeed, a single environment cannot allow identifying such attacks.

- Random Propagation Activities: This is the second category of Type I attacks we can imagine. Many tools choose random IPs during their propagation process. This can be worms or bots (Sasser, W32/Agobot, Bobax, etc... [Res04, SOP04]). As they choose their victims randomly, it is very likely we observe the same Source only once. Indeed, we notice that IP Addresses in general are not observed twice on both environments H_1 and on H_2. To identify these Type I attacks, we have decided to build a technique upon the work already published: we have presented in [PD04a] a clustering algorithm that allows identifying root causes of frequent processes observed in one environment. Due to space limitations, we report the interested reader to [PD04a] for a detailed description of the clustering technique. To make it short, we basically gather all attacks presenting some common features (duration of the attacks, number of packets sent, targeted ports...) based on generalization techniques and association-rules mining. The resulting clusters are further refined using "phrase distance" between attack payloads. In summary, we gather on one cluster attack sources having performed same attack processes and which have many common characteristics.
  As a consequence, tools propagating through random IPs have similar characteristics, even if they are not observed twice on the environments, so they should belong to the very same cluster. These Type I sources are more precisely characterized by clusters where all Sources have targeted only one virtual machine, and where the attacks within a same cluster are equally distributed among virtual machines. If the distribution of the attacks per virtual machine is equiprobable (which means we do not observe a significant number of attacks on a limited number of honeypots), we consider the associated attack belongs to this category. We perform it in an automatic way following the algorithm presented in Figure 3.3 for each cluster. It simply consists in finding all clusters that have the property explained above. A result of 1 means that the attack associated to Cluster C_j belongs to the *Random Propagation Attack category.* Null Results are discussed in the following category.
  If we consider the 240 clusters associated to attacks on H_1, only 54 correspond to type I attacks. In addition, 43 out of these 54 clusters have *random propagation strategies* (according to algorithm 3.3).The remaining 0.5% of the observed clusters that are associated to type I attacks are discussed in the next category. Finally, we want to point out that attacks on that category can be identified from either platform H_1 or H_2.
- Targeted Attacks: This is the third category of Type I attacks. It gathers all Type I attacks which cannot be classified in the two previous categories.

They are not numerous, as explained above. They are represented by 0.5% of the clusters and imply a few dozens of attacking Sources. This category regroups various attacks of interest, due to their originality. These attacks have targeted always the same virtual machine only one environment. The reason why some attacks focus on one machine only are really worth analyzing to determine if a specific service is targeted or if it is due to other phenomena. In the following, we give two illustrative examples:

- *Example 1: Attacks on port 25666 targeting virtual machine mach0 on* H_1*.* This attack has been observed 387 times from 378 different IP addresses between August 2004 and February 2005. Each attack source sends in average three packets to mach0. A closer look reveals that all packets have 80 or 8080 (http) as TCP source port and RST-ACK flags set. They are replies to DoS attacks against web servers, also known as *backscatters* ([MVS01]). In summary, we have observed for 6 months DoS attacks against different web servers, and these attacks always spoofed mach0 IP address with source port 25666. Such regular processes have been observed in both German and French environments. Up to now, we observed 15 of these processes.

  First, the attacks are very steady over time. We observe at least traces every day in the given example. Second, it seems surprising DoS tools spoof static addresses: either spoofed (IP,port) are somehow hardcoded in the tool (which would be more than bizarre), or this six-month DoS attacks are part of a unique process launched against them over months. By a unique process, we suggest that the spoofed address list has been generated once, and it has then been used for multiple attacks. The regularity of such a process also indicates a common cause that has decided to target quite a stable amount of servers each day. Finally, these *periodic backscatters* come to ports that are likely close on both environments (hugely very high non-privileged ports in the range $[1025, 65535]$). Thus, we would get the same amount of information, whatever the targeted environment is.

- *Example 2: Targeted port 5000 Attack on mach1 on* H_2*.* Two very different worms are currently responsible for port 5000 scans. The first, *Bobax*, uses port 5000 to identify Windows XP systems. Windows XP uses port 5000 (TCP) for 'Universal Plug and Play (UPnP)', which is by default open. The second worm, *Kibuv*, uses an old vulnerability in Windows XP's UPnP implementation to exploit systems. This vulnerability was one of the first discovered in Windows XP and patches have been available. However, we observe a cluster that is associated to that port. It gathers 73 distinct IP sources that have targeted only one honeypot machine on port sequence 5000. Surprisingly enough, the 73 attacks occurred on the very same virtual machine within two months. This does not match the Bobax and Kibuv worm propagation scheme, as it has been found they scan machines randomly. In addition, it is important to note that the port is closed on that machine. Packets contain no payload. They are limited to half a dozen TCP SYN packets. This attack cannot be considered as random insofar as it always implies the same virtual target.

At this time writing, we have no concrete explanation of such a phenomenon. It has also been noticed by other administrators in Incidents mailing lists [san04]. The Michigan Internet Motion Sensors group notifies in [ins04] that the observed activities do "not support the theory of Kibuv entirely". This might be due to revived threats such as *Sockets de Troie (Blazer 5)* or *1998 Trojan ICKiller* or Yahoo Chat or non-referenced tools based on the UPnP exploit [ick05, upn03]. A closer look at the received packets is required at this stage to determine the attack. However, as the port 5000 is close in both platforms H_1 and H_2, we would get the same amount of information, whatever the targeted environment is.

For each Cluster C_j of type I:

Preliminaries :

Compute the number N_j of attacks associated to C_j on the Environment
Compute the number $N_{j,0}$ of attacks associated to C_j on the virtual machine mach0
Compute the number $N_{j,1}$ of attacks associated to C_j on the virtual machine mach1
Compute the number $N_{j,2}$ of attacks associated to C_j on the virtual machine mach2
We check that $N_{j,0} + N_{j,1} + N_{j,2} = N_j$
$Threshold = 0.1 N_j$

Test on Cluster C_j:

Mean $= \mu = \frac{N_j}{3}$

variance $= \sigma^2 = \frac{\sum_{0 \le k \le 2} (N_{j,k} - \mu)^2}{3}$

IF $\sigma < Threshold$
THEN
$res = 1$
Cluster C_j associated to random propagation tools
ELSE
$res = 0$
Cluster C_j associated to targeted attacks
A closer look at packet contents is required.

Table 3. Verification process of a request with the improved protocol

Type I attacks are very interesting. After distinguishing backscatters and tools with widespread random propagation, a few numbers of attacks remain unclassified. They seem to be specific to the platform itself, so some precautions must be required to understand them. At this time writing, they are hidden in the noisy permanent activities and thus, they do not really trigger lots of attention. Simple honeypots emulating a few IPs allow identifying them. This is a preliminary but necessary step to start their in-depth analysis. Then, more interaction on that port would bring valuable information on that attack. As the attack is very specific and we have no preliminary knowledge on it, writing a simple script to H_2 is not the correct choice. A controlled environment like H_1 must be built to observe the attack details against real interactive systems. In a second step, a script can be developed for H_2.

We show here that high interaction honeypots are very complementary to low interaction honeypots as they can indicate which services are not currently interactive enough on low interaction honeypots. We intend in the last Section to make this analysis more automatic so that we can determine which services must be developed (by means of scripts) on the low interaction honeypot to get a similar amount of information.

Interaction Differences and Improvements The platforms are globally targeted the same way, as it has been detailed in the last Sections. However, it is also clear that we collect more data on a high interaction honeypot, as real services are communicating. In average, 50 more times packets are collected with H_1 than with H_2. Based on these observations, this Section intends to show where the information is lacking, and how this can be handled.

As specified in Section 2, platforms H_1 and H_2 have similar configurations. All open ports on machines in H_1 are also opened in H_2, and vice-versa. On H_2 side, it can be sufficient to open a port in order to get attack information. It can also be necessary to develop simple emulation scripts in order to enhance the environment interaction. Thus, the idea is the following: The more attacks interact with a port, the more important it is that Honeyd runs an interactive script behind. In other words, if the amount of information we obtain on attacks through a given port on H_1 is a lot higher than the one captured on H_2 against the same port, one of the two following actions must be undertaken:

- A script must be implemented to emulate the associated service if any.
- The script interaction should be brought to a higher level if the script already exists.

Obviously enough, each attack may require different interaction levels. For instance, scans do not require high interaction and an open port on both environments will give the same amount of information.

Furthermore, the error would be here to consider only packets from/to a given port to compare the amount of information between the two environments. For instance, if a Source sends a request on port A and then waits for the answer to communicate with port B, the lacking information if port A is closed on the other environment is a lot more important than just considering the simple request/answer on port A. We miss all the communication with port B either.

As a consequence, we use the clusters presented in [PD04a] and introduced in Section 3.3 to avoid these problems and to determine what services are not correctly interactive on H_2. Each cluster groups together all IP Sources sharing strong characteristics on their attack processes. These attacking sources have exchanged quite the same amount of information on one environment. The interaction we get on a virtual machine must be weighted by the frequency of the attacks on the involved ports, as we explain above. The interaction is quantified by considering the number of exchanged packets. This can be refined by taking payload length into account, but we limit this analysis on this simple assumption. This leads to the following algorithm in Figure 4:

The algorithm has been launched on each platform for a 2-month period. We get the following results:

- For ports where simple scripts are already attached to in H_2, it appears they behave correctly compared to the real services running in H_1.

Preliminaries :

For Successive Environments H_1 and H_2:
For each Virtual Machine M_j and each associated port $p_{j,k}$:

Gather the list of Clusters $C_{l,k}$ corresponding to attacks on Virtual Machine M_j against at least port $p_{j,k}$
Be N the total number of IP Sources having targeted Virtual machine M_j
Be η the threshold to compare interactions between environments. $\eta = 0.7$
FOR EACH Cluster $C_{l,k}$
Compute the number n_l of Sources belonging to Cluster $C_{l,k}$
Compute P_l, the total number of exchanged packets between Sources belonging to Cluster $C_{l,k}$
Compute the *frequency* of Cluster $C_{l,k}$ as

$f_l = \frac{n_l}{N}$

Interaction Estimation:

The interaction estimation is for H_1

$I(H_1) = \sum_{l \geq 1} P_l . f_l$

The interaction estimation is for H_2

$I(H_2) = \sum_{m \geq 1} P_m . f_m$

Analysis:

IF $\frac{I(H_2)}{I(H_1)} \leq \eta$
The current implementation on port $p_{j,k}$ for Virtual Machine M_j in H_2 is not correct
The Interaction on this port is not satisfactory. The associated script should be enhanced.

Table 4. Comparing Interactions between H_1 and H_2

- For ports Microsoft and Netbios (135, 139 and 445 specially), the ratio $\frac{I(H_2)}{I(H_1)}$ is equal to 1.5%. No script emulates these services in H_2. This is clearly not acceptable, insofar as H_2 is missing a large quantity of information in comparison to H_1. We are in the process of writing scripts to emulate these services.
- For other ports like 111, 515,..., the operation of opening these ports provides as much information as the real services in H_1 at this time. There is no need to emulate these services.

The algorithm gives an important hint of which ports are not correctly configured on the low interaction environment. It also provides a priority list of those emulating ports that should be urgently modified. The result confirms that most of the missing information comes from the Microsoft services. To conclude, this algorithm highlights the important complementarities that can be obtained by using both a high interaction and a low interaction honeypot.

4 Leurre.com Project

4.1 Motivations

We have presented in previous publications some experiments based on a high interaction honeypot [PD04a, PD04b]. These experiments have shown that most of the attacks are caused by a few number of attack tools and that there are very stable processes occurring in the wild. As one major objective is to get statistical information from the attacks, it appears that lower interaction honeypots can be sufficient enough. Indeed, we do not want our platforms to be corrupted as main honeypot applications intend to [Spi02b]. We only want to observe the first attack steps in order to get a better understanding of current malicious activities. This paper provides another strong motivation, as it shows that low interaction honeypots brings equivalent global statistics on the attacks. In addition, some regular comparisons like we do between both types of environments lead to an optimization of the low interaction configuration.

As a direct consequence, we have decided to deploy low interaction honeypots in various places. Leurre.com project aims at disseminating such platforms everywhere thanks to motivated partners as part of the Leurre.com project. Partners are invited to join this project and install one platform on their own. Eurecom takes care of the installation by furnishing the platform image and configuration files. Thus, the install process is automatic. In exchange, we give the partners an access to the database and its enriched information [5]. We are also developing a dedicated web to make research faster and more efficient. The project starts triggering interest from many organizations, whatever academic, industrial of governmental. We hope the number of partners will keep on increasing in a near future.

5 Conclusion

This paper presents a very important contribution to the Leurre.com Project. Indeed, it shows on one hand that high interaction honeypots are somehow superfluous in the large-scale deployment of statistical sensors, since global statistics remain very similar. On the other hand, it shows that they are vital to control the configuration relevance of low interaction honeypots. This leads to the conclusion that complementarities between high and low interaction honeypots can increase the accuracy of information collected by simple environments deployed in different places.

Besides, this comparison has led to an interesting analysis of collected data. First, it allows identifying very specific attacks and weird phenomena, as this has been shown through some examples. The latter require particular attention to be analysed and understood. Second, it highlights the need to take into account packet losses in the analysis of malicious data. Otherwise, this can lead to many misunderstandings. Most of the previous points must be carefully analyzed and are part of our future work.

Finally, we hope this paper will be an incitement for other partners to join the open project Leurre.com that we are deploying.

References

[adv05] The AdvanceSCAN advscan utility, 2005. URL: http://advancemame.sourceforge.net/doc-advscan.html.

[Cai05] CAIDA, the Cooperative Association for Internet Data Analysis. Internet: `http://www.caida.org/`, 2005.

[CBM+04] E. Cooke, M. Bailey, Z.M. Mao, D. Watson, F. Jahanian, and D. McPherson. Toward understanding distributed blackhole placement. In *Proceedings of the Recent Advances of Intrusion Detection RAID'04*, September 2004.

[5] A Non-Disclosure Agreement is signed to protect partners from each others

[Cen] The SANS Institute Internet Storm Center. The trusted source for computer security trainind, certification and research. URL:http://isc.sans.org.

[Cen01] Stanford Linear Accelerator Center. Tutorial on internet monitoring and pinger, 2001. URL: http://www.slac.stanford.edu/comp/net/wan-mon/tutorial.html.

[CM] B. Gemberling C. Morrow. How to allow your customers to blackhole their own traffic. URL:http://www.secsup.org/CustomerBlackhole/.

[Cor] VMWare Corporation. User' s manual. version 4.1. URL:http://www.vmware.com.

[Cym04] Team Cymru: The Darknet Project. Internet: `http://www.cymru.com/Darknet/`, 2004.

[DPD04a] M. Dacier, F. Pouget, and H. Debar. Attack processes found on the internet. In *NATO Symposium IST-041/RSY-013*, April 2004.

[DPD04b] M. Dacier, F. Pouget, and H. Debar. Honeypots, a practical mean to validate malicious fault assumptions. In *The 10th Pacific Ream Dependable Computing Conference (PRDC04)*, February 2004.

[hon04] honeyd Homepage. Internet: `http://honeyd.org/`, 2004.

[ick05] Security Port Scanner, Trojan Port List: ICKiller. Internet: `http://www.glocksoft.com/trojan_list/ICKiller.htm`, 2005.

[ins04] TCP port 5000 syn increasing. Internet: `http://seclists.org/lists/incidents/2004/May/0074.html`, 2004.

[LUR04] LURHQ. Dabber worm analysis, 2004. URL: http://www.lurhq.com/dabber.html.

[max04] MaxMind: Geolocation and Credit Card Fraud Detection. Internet: `http://www.maxmind.com`, 2004.

[MVS01] D. Moore, G. Voelker, and S. Savage. Infering internet denial-of-service activity. In *The USENIX Security Symposium*, August 2001.

[PD04a] F. Pouget and M. Dacier. Honeypot-based forensics. In *AusCERT Asia Pacific Information Technology Security Conference 2004 (AusCERT2004)*, May 2004.

[PD04b] Fabien Pouget and Marc Dacier. Honeypot-based forensics. In George Mohay, Andrew Clark, and Kathryn Kerr, editors, *Proceedings of AusCERT Asia Pacific Information Technology Security Conference 2004*, pages 1–15, 2004.

[PDD05] F. Pouget, M. Dacier, and H. Debar. Honeynets: Foundations for the development of early warning systems. 2005. Publisher Springler-Verlag, LNCS, NATO ARW Series.

[pof04] p0f: Passive OS Fingerprinting Tool. Internet: `http://lcamtuf.coredump.cx/p0f.shtml`, 2004.

[Rep05] Internet Traffic Report, 2005. URL: http://www.internettrafficreport.com/main.htm.

[Res04] Symantec Security Response. W32-sasser.worm, 2004. URL: http://securityresponse.symantec.com/ avcenter/venc/data/w32.sasser.worm.html.

[san04] 5000 spike? Internet: `http://lists.sans.org/pipermail/list/2004-May/048192.html`, 2004.

[SMS] D. Song, R. Malan, and R. Stone. A global snapshot of internet worm activity. Technical report. URL:http://research.arbor.net/downloads/snapshot_worm_activity.pdf.

[SOP04] SOPHOS. Sophos virus analysis: W32/agobot-pq, 2004. URL: http://www.sophos.com.au/virusinfo/analyses/w32agobotpq.html.

[Spi02a] L. Spitzner. *Honeypots: Tracking Hackers*. Addison-Wesley, 2002.

[Spi02b] L. Spitzner. *Honeypots: Tracking Hackers*. Addison Wesley, 2002.

[Sys] Cisco Systems. Netflow Services and Applications (1999).

[The03] The Honeynet Project. Know Your Enemy: GenII Honeynets, 2003. `http://www.honeynet.org/papers/gen2/`.

[upn03] 2003 UPnP Exploit. Internet: `http://www.packetstormsecurity.org/0112-exploits/XPloit.c`, 2003.

[uti] TCPDump utility. URL:http://www.tcpdump.org.

[YBJ04] V. Yegneswaran, P. Barford, and S. Jha. Global intrusion detection in the domino overlay system. 2004.

Aachener Informatik-Berichte

This is a list of recent technical reports. To obtain copies of technical reports please consult `http://aib.informatik.rwth-aachen.de/` or send your request to: Informatik-Bibliothek, RWTH Aachen, Ahornstr. 55, 52056 Aachen, Email: `biblio@informatik.rwth-aachen.de`

1987-01 *	Fachgruppe Informatik: Jahresbericht 1986
1987-02 *	David de Frutos Escrig, Klaus Indermark: Equivalence Relations of Non-Deterministic Ianov-Schemes
1987-03 *	Manfred Nagl: A Software Development Environment based on Graph Technology
1987-04 *	Claus Lewerentz, Manfred Nagl, Bernhard Westfechtel: On Integration Mechanisms within a Graph-Based Software Development Environment
1987-05 *	Reinhard Rinn: Über Eingabeanomalien bei verschiedenen Inferenzmodellen
1987-06 *	Werner Damm, Gert Döhmen: Specifying Distributed Computer Architectures in AADL*
1987-07 *	Gregor Engels, Claus Lewerentz, Wilhelm Schäfer: Graph Grammar Engineering: A Software Specification Method
1987-08 *	Manfred Nagl: Set Theoretic Approaches to Graph Grammars
1987-09 *	Claus Lewerentz, Andreas Schürr: Experiences with a Database System for Software Documents
1987-10 *	Herbert Klaeren, Klaus Indermark: A New Implementation Technique for Recursive Function Definitions
1987-11 *	Rita Loogen: Design of a Parallel Programmable Graph Reduction Machine with Distributed Memory
1987-12	J. Börstler, U. Möncke, R. Wilhelm: Table compression for tree automata
1988-01 *	Gabriele Esser, Johannes Rückert, Frank Wagner: Gesellschaftliche Aspekte der Informatik
1988-02 *	Peter Martini, Otto Spaniol: Token-Passing in High-Speed Backbone Networks for Campus-Wide Environments
1988-03 *	Thomas Welzel: Simulation of a Multiple Token Ring Backbone
1988-04 *	Peter Martini: Performance Comparison for HSLAN Media Access Protocols
1988-05 *	Peter Martini: Performance Analysis of Multiple Token Rings
1988-06 *	Andreas Mann, Johannes Rückert, Otto Spaniol: Datenfunknetze
1988-07 *	Andreas Mann, Johannes Rückert: Packet Radio Networks for Data Exchange
1988-08 *	Andreas Mann, Johannes Rückert: Concurrent Slot Assignment Protocol for Packet Radio Networks
1988-09 *	W. Kremer, F. Reichert, J. Rückert, A. Mann: Entwurf einer Netzwerktopologie für ein Mobilfunknetz zur Unterstützung des öffentlichen Straßenverkehrs
1988-10 *	Kai Jakobs: Towards User-Friendly Networking
1988-11 *	Kai Jakobs: The Directory - Evolution of a Standard
1988-12 *	Kai Jakobs: Directory Services in Distributed Systems - A Survey
1988-13 *	Martine Schümmer: RS-511, a Protocol for the Plant Floor

1988-14 * U. Quernheim: Satellite Communication Protocols - A Performance Comparison Considering On-Board Processing

1988-15 * Peter Martini, Otto Spaniol, Thomas Welzel: File Transfer in High Speed Token Ring Networks: Performance Evaluation by Approximate Analysis and Simulation

1988-16 * Fachgruppe Informatik: Jahresbericht 1987

1988-17 * Wolfgang Thomas: Automata on Infinite Objects

1988-18 * Michael Sonnenschein: On Petri Nets and Data Flow Graphs

1988-19 * Heiko Vogler: Functional Distribution of the Contextual Analysis in Block-Structured Programming Languages: A Case Study of Tree Transducers

1988-20 * Thomas Welzel: Einsatz des Simulationswerkzeuges QNAP2 zur Leistungsbewertung von Kommunikationsprotokollen

1988-21 * Th. Janning, C. Lewerentz: Integrated Project Team Management in a Software Development Environment

1988-22 * Joost Engelfriet, Heiko Vogler: Modular Tree Transducers

1988-23 * Wolfgang Thomas: Automata and Quantifier Hierarchies

1988-24 * Uschi Heuter: Generalized Definite Tree Languages

1989-01 * Fachgruppe Informatik: Jahresbericht 1988

1989-02 * G. Esser, J. Rückert, F. Wagner (Hrsg.): Gesellschaftliche Aspekte der Informatik

1989-03 * Heiko Vogler: Bottom-Up Computation of Primitive Recursive Tree Functions

1989-04 * Andy Schürr: Introduction to PROGRESS, an Attribute Graph Grammar Based Specification Language

1989-05 J. Börstler: Reuse and Software Development - Problems, Solutions, and Bibliography (in German)

1989-06 * Kai Jakobs: OSI - An Appropriate Basis for Group Communication?

1989-07 * Kai Jakobs: ISO's Directory Proposal - Evolution, Current Status and Future Problems

1989-08 * Bernhard Westfechtel: Extension of a Graph Storage for Software Documents with Primitives for Undo/Redo and Revision Control

1989-09 * Peter Martini: High Speed Local Area Networks - A Tutorial

1989-10 * P. Davids, Th. Welzel: Performance Analysis of DQDB Based on Simulation

1989-11 * Manfred Nagl (Ed.): Abstracts of Talks presented at the WG '89 15th International Workshop on Graphtheoretic Concepts in Computer Science

1989-12 * Peter Martini: The DQDB Protocol - Is it Playing the Game?

1989-13 * Martine Schümmer: CNC/DNC Communication with MAP

1989-14 * Martine Schümmer: Local Area Networks for Manufactoring Environments with hard Real-Time Requirements

1989-15 * M. Schümmer, Th. Welzel, P. Martini: Integration of Field Bus and MAP Networks - Hierarchical Communication Systems in Production Environments

1989-16 * G. Vossen, K.-U. Witt: SUXESS: Towards a Sound Unification of Extensions of the Relational Data Model

1989-17 * J. Derissen, P. Hruschka, M.v.d. Beeck, Th. Janning, M. Nagl: Integrating Structured Analysis and Information Modelling
1989-18 A. Maassen: Programming with Higher Order Functions
1989-19 * Mario Rodriguez-Artalejo, Heiko Vogler: A Narrowing Machine for Syntax Directed BABEL
1989-20 H. Kuchen, R. Loogen, J.J. Moreno Navarro, M. Rodriguez Artalejo: Graph-based Implementation of a Functional Logic Language
1990-01 * Fachgruppe Informatik: Jahresbericht 1989
1990-02 * Vera Jansen, Andreas Potthoff, Wolfgang Thomas, Udo Wermuth: A Short Guide to the AMORE System (Computing Automata, MOnoids and Regular Expressions)
1990-03 * Jerzy Skurczynski: On Three Hierarchies of Weak SkS Formulas
1990-04 R. Loogen: Stack-based Implementation of Narrowing
1990-05 H. Kuchen, A. Wagener: Comparison of Dynamic Load Balancing Strategies
1990-06 * Kai Jakobs, Frank Reichert: Directory Services for Mobile Communication
1990-07 * Kai Jakobs: What's Beyond the Interface - OSI Networks to Support Cooperative Work
1990-08 * Kai Jakobs: Directory Names and Schema - An Evaluation
1990-09 * Ulrich Quernheim, Dieter Kreuer: Das CCITT - Signalisierungssystem Nr. 7 auf Satellitenstrecken; Simulation der Zeichengabestrecke
1990-11 H. Kuchen, R. Loogen, J.J. Moreno Navarro, M. Rodriguez Artalejo: Lazy Narrowing in a Graph Machine
1990-12 * Kai Jakobs, Josef Kaltwasser, Frank Reichert, Otto Spaniol: Der Computer fährt mit
1990-13 * Rudolf Mathar, Andreas Mann: Analyzing a Distributed Slot Assignment Protocol by Markov Chains
1990-14 A. Maassen: Compilerentwicklung in Miranda - ein Praktikum in funktionaler Programmierung (written in german)
1990-15 * Manfred Nagl, Andreas Schürr: A Specification Environment for Graph Grammars
1990-16 A. Schürr: PROGRESS: A VHL-Language Based on Graph Grammars
1990-17 * Marita Möller: Ein Ebenenmodell wissensbasierter Konsultationen - Unterstützung für Wissensakquisition und Erklärungsfähigkeit
1990-18 * Eric Kowalewski: Entwurf und Interpretation einer Sprache zur Beschreibung von Konsultationsphasen in Expertensystemen
1990-20 Y. Ortega Mallen, D. de Frutos Escrig: A Complete Proof System for Timed Observations
1990-21 * Manfred Nagl: Modelling of Software Architectures: Importance, Notions, Experiences
1990-22 H. Fassbender, H. Vogler: A Call-by-need Implementation of Syntax Directed Functional Programming
1991-01 Guenther Geiler (ed.), Fachgruppe Informatik: Jahresbericht 1990
1991-03 B. Steffen, A. Ingolfsdottir: Characteristic Formulae for Processes with Divergence
1991-04 M. Portz: A new class of cryptosystems based on interconnection networks

1991-05 H. Kuchen, G. Geiler: Distributed Applicative Arrays
1991-06 * Ludwig Staiger: Kolmogorov Complexity and Hausdorff Dimension
1991-07 * Ludwig Staiger: Syntactic Congruences for w-languages
1991-09 * Eila Kuikka: A Proposal for a Syntax-Directed Text Processing System
1991-10 K. Gladitz, H. Fassbender, H. Vogler: Compiler-based Implementation of Syntax-Directed Functional Programming
1991-11 R. Loogen, St. Winkler: Dynamic Detection of Determinism in Functional Logic Languages
1991-12 * K. Indermark, M. Rodriguez Artalejo (Eds.): Granada Workshop on the Integration of Functional and Logic Programming
1991-13 * Rolf Hager, Wolfgang Kremer: The Adaptive Priority Scheduler: A More Fair Priority Service Discipline
1991-14 * Andreas Fasbender, Wolfgang Kremer: A New Approximation Algorithm for Tandem Networks with Priority Nodes
1991-15 J. Börstler, A. Zündorf: Revisiting extensions to Modula-2 to support reusability
1991-16 J. Börstler, Th. Janning: Bridging the gap between Requirements Analysis and Design
1991-17 A. Zündorf, A. Schürr: Nondeterministic Control Structures for Graph Rewriting Systems
1991-18 * Matthias Jarke, John Mylopoulos, Joachim W. Schmidt, Yannis Vassiliou: DAIDA: An Environment for Evolving Information Systems
1991-19 M. Jeusfeld, M. Jarke: From Relational to Object-Oriented Integrity Simplification
1991-20 G. Hogen, A. Kindler, R. Loogen: Automatic Parallelization of Lazy Functional Programs
1991-21 * Prof. Dr. rer. nat. Otto Spaniol: ODP (Open Distributed Processing): Yet another Viewpoint
1991-22 H. Kuchen, F. Lücking, H. Stoltze: The Topology Description Language TDL
1991-23 S. Graf, B. Steffen: Compositional Minimization of Finite State Systems
1991-24 R. Cleaveland, J. Parrow, B. Steffen: The Concurrency Workbench: A Semantics Based Tool for the Verification of Concurrent Systems
1991-25 * Rudolf Mathar, Jürgen Mattfeldt: Optimal Transmission Ranges for Mobile Communication in Linear Multihop Packet Radio Networks
1991-26 M. Jeusfeld, M. Staudt: Query Optimization in Deductive Object Bases
1991-27 J. Knoop, B. Steffen: The Interprocedural Coincidence Theorem
1991-28 J. Knoop, B. Steffen: Unifying Strength Reduction and Semantic Code Motion
1991-30 T. Margaria: First-Order theories for the verification of complex FSMs
1991-31 B. Steffen: Generating Data Flow Analysis Algorithms from Modal Specifications
1992-01 Stefan Eherer (ed.), Fachgruppe Informatik: Jahresbericht 1991
1992-02 * Bernhard Westfechtel: Basismechanismen zur Datenverwaltung in strukturbezogenen Hypertextsystemen
1992-04 S. A. Smolka, B. Steffen: Priority as Extremal Probability
1992-05 * Matthias Jarke, Carlos Maltzahn, Thomas Rose: Sharing Processes: Team Coordination in Design Repositories

1992-06 O. Burkart, B. Steffen: Model Checking for Context-Free Processes

1992-07 * Matthias Jarke, Klaus Pohl: Information Systems Quality and Quality Information Systems

1992-08 * Rudolf Mathar, Jürgen Mattfeldt: Analyzing Routing Strategy NFP in Multihop Packet Radio Networks on a Line

1992-09 * Alfons Kemper, Guido Moerkotte: Grundlagen objektorientierter Datenbanksysteme

1992-10 Matthias Jarke, Manfred Jeusfeld, Andreas Miethsam, Michael Gocek: Towards a logic-based reconstruction of software configuration management

1992-11 Werner Hans: A Complete Indexing Scheme for WAM-based Abstract Machines

1992-12 W. Hans, R. Loogen, St. Winkler: On the Interaction of Lazy Evaluation and Backtracking

1992-13 * Matthias Jarke, Thomas Rose: Specification Management with CAD

1992-14 Th. Noll, H. Vogler: Top-down Parsing with Simultaneous Evaluation on Noncircular Attribute Grammars

1992-15 A. Schuerr, B. Westfechtel: Graphgrammatiken und Graphersetzungssysteme(written in german)

1992-16 * Graduiertenkolleg Informatik und Technik (Hrsg.): Forschungsprojekte des Graduiertenkollegs Informatik und Technik

1992-17 M. Jarke (ed.): ConceptBase V3.1 User Manual

1992-18 * Clarence A. Ellis, Matthias Jarke (Eds.): Distributed Cooperation in Integrated Information Systems - Proceedings of the Third International Workshop on Intelligent and Cooperative Information Systems

1992-19-00 H. Kuchen, R. Loogen (eds.): Proceedings of the 4th Int. Workshop on the Parallel Implementation of Functional Languages

1992-19-01 G. Hogen, R. Loogen: PASTEL - A Parallel Stack-Based Implementation of Eager Functional Programs with Lazy Data Structures (Extended Abstract)

1992-19-02 H. Kuchen, K. Gladitz: Implementing Bags on a Shared Memory MIMD-Machine

1992-19-03 C. Rathsack, S.B. Scholz: LISA - A Lazy Interpreter for a Full-Fledged Lambda-Calculus

1992-19-04 T.A. Bratvold: Determining Useful Parallelism in Higher Order Functions

1992-19-05 S. Kahrs: Polymorphic Type Checking by Interpretation of Code

1992-19-06 M. Chakravarty, M. Köhler: Equational Constraints, Residuation, and the Parallel JUMP-Machine

1992-19-07 J. Seward: Polymorphic Strictness Analysis using Frontiers (Draft Version)

1992-19-08 D. Gärtner, A. Kimms, W. Kluge: pi-Red^+ - A Compiling Graph-Reduction System for a Full Fledged Lambda-Calculus

1992-19-09 D. Howe, G. Burn: Experiments with strict STG code

1992-19-10 J. Glauert: Parallel Implementation of Functional Languages Using Small Processes

1992-19-11 M. Joy, T. Axford: A Parallel Graph Reduction Machine

1992-19-12 A. Bennett, P. Kelly: Simulation of Multicache Parallel Reduction

1992-19-13 K. Langendoen, D.J. Agterkamp: Cache Behaviour of Lazy Functional Programs (Working Paper)

1992-19-14 K. Hammond, S. Peyton Jones: Profiling scheduling strategies on the GRIP parallel reducer

1992-19-15 S. Mintchev: Using Strictness Information in the STG-machine

1992-19-16 D. Rushall: An Attribute Grammar Evaluator in Haskell

1992-19-17 J. Wild, H. Glaser, P. Hartel: Statistics on storage management in a lazy functional language implementation

1992-19-18 W.S. Martins: Parallel Implementations of Functional Languages

1992-19-19 D. Lester: Distributed Garbage Collection of Cyclic Structures (Draft version)

1992-19-20 J.C. Glas, R.F.H. Hofman, W.G. Vree: Parallelization of Branch-and-Bound Algorithms in a Functional Programming Environment

1992-19-21 S. Hwang, D. Rushall: The nu-STG machine: a parallelized Spineless Tagless Graph Reduction Machine in a distributed memory architecture (Draft version)

1992-19-22 G. Burn, D. Le Metayer: Cps-Translation and the Correctness of Optimising Compilers

1992-19-23 S.L. Peyton Jones, P. Wadler: Imperative functional programming (Brief summary)

1992-19-24 W. Damm, F. Liu, Th. Peikenkamp: Evaluation and Parallelization of Functions in Functional + Logic Languages (abstract)

1992-19-25 M. Kesseler: Communication Issues Regarding Parallel Functional Graph Rewriting

1992-19-26 Th. Peikenkamp: Charakterizing and representing neededness in functional loginc languages (abstract)

1992-19-27 H. Doerr: Monitoring with Graph-Grammars as formal operational Models

1992-19-28 J. van Groningen: Some implementation aspects of Concurrent Clean on distributed memory architectures

1992-19-29 G. Ostheimer: Load Bounding for Implicit Parallelism (abstract)

1992-20 H. Kuchen, F.J. Lopez Fraguas, J.J. Moreno Navarro, M. Rodriguez Artalejo: Implementing Disequality in a Lazy Functional Logic Language

1992-21 H. Kuchen, F.J. Lopez Fraguas: Result Directed Computing in a Functional Logic Language

1992-22 H. Kuchen, J.J. Moreno Navarro, M.V. Hermenegildo: Independent AND-Parallel Narrowing

1992-23 T. Margaria, B. Steffen: Distinguishing Formulas for Free

1992-24 K. Pohl: The Three Dimensions of Requirements Engineering

1992-25 * R. Stainov: A Dynamic Configuration Facility for Multimedia Communications

1992-26 * Michael von der Beeck: Integration of Structured Analysis and Timed Statecharts for Real-Time and Concurrency Specification

1992-27 W. Hans, St. Winkler: Aliasing and Groundness Analysis of Logic Programs through Abstract Interpretation and its Safety

1992-28 * Gerhard Steinke, Matthias Jarke: Support for Security Modeling in Information Systems Design

1992-29 B. Schinzel: Warum Frauenforschung in Naturwissenschaft und Technik

1992-30 A. Kemper, G. Moerkotte, K. Peithner: Object-Orientation Axiomatised by Dynamic Logic

1992-32 * Bernd Heinrichs, Kai Jakobs: Timer Handling in High-Performance Transport Systems

1992-33 * B. Heinrichs, K. Jakobs, K. Lenßen, W. Reinhardt, A. Spinner: Euro-Bridge: Communication Services for Multimedia Applications

1992-34 C. Gerlhof, A. Kemper, Ch. Kilger, G. Moerkotte: Partition-Based Clustering in Object Bases: From Theory to Practice

1992-35 J. Börstler: Feature-Oriented Classification and Reuse in IPSEN

1992-36 M. Jarke, J. Bubenko, C. Rolland, A. Sutcliffe, Y. Vassiliou: Theories Underlying Requirements Engineering: An Overview of NATURE at Genesis

1992-37 * K. Pohl, M. Jarke: Quality Information Systems: Repository Support for Evolving Process Models

1992-38 A. Zuendorf: Implementation of the imperative / rule based language PROGRES

1992-39 P. Koch: Intelligentes Backtracking bei der Auswertung funktional-logischer Programme

1992-40 * Rudolf Mathar, Jürgen Mattfeldt: Channel Assignment in Cellular Radio Networks

1992-41 * Gerhard Friedrich, Wolfgang Neidl: Constructive Utility in Model-Based Diagnosis Repair Systems

1992-42 * P. S. Chen, R. Hennicker, M. Jarke: On the Retrieval of Reusable Software Components

1992-43 W. Hans, St.Winkler: Abstract Interpretation of Functional Logic Languages

1992-44 N. Kiesel, A. Schuerr, B. Westfechtel: Design and Evaluation of GRAS, a Graph-Oriented Database System for Engineering Applications

1993-01 * Fachgruppe Informatik: Jahresbericht 1992

1993-02 * Patrick Shicheng Chen: On Inference Rules of Logic-Based Information Retrieval Systems

1993-03 G. Hogen, R. Loogen: A New Stack Technique for the Management of Runtime Structures in Distributed Environments

1993-05 A. Zündorf: A Heuristic for the Subgraph Isomorphism Problem in Executing PROGRES

1993-06 A. Kemper, D. Kossmann: Adaptable Pointer Swizzling Strategies in Object Bases: Design, Realization, and Quantitative Analysis

1993-07 * Graduiertenkolleg Informatik und Technik (Hrsg.): Graduiertenkolleg Informatik und Technik

1993-08 * Matthias Berger: k-Coloring Vertices using a Neural Network with Convergence to Valid Solutions

1993-09 M. Buchheit, M. Jeusfeld, W. Nutt, M. Staudt: Subsumption between Queries to Object-Oriented Databases

1993-10 O. Burkart, B. Steffen: Pushdown Processes: Parallel Composition and Model Checking

1993-11 * R. Große-Wienker, O. Hermanns, D. Menzenbach, A. Pollacks, S. Repetzki, J. Schwartz, K. Sonnenschein, B. Westfechtel: Das SUKITS-Projekt: A-posteriori-Integration heterogener CIM-Anwendungssysteme

1993-12 * Rudolf Mathar, Jürgen Mattfeldt: On the Distribution of Cumulated Interference Power in Rayleigh Fading Channels
1993-13 O. Maler, L. Staiger: On Syntactic Congruences for omega-languages
1993-14 M. Jarke, St. Eherer, R. Gallersdoerfer, M. Jeusfeld, M. Staudt: ConceptBase - A Deductive Object Base Manager
1993-15 M. Staudt, H.W. Nissen, M.A. Jeusfeld: Query by Class, Rule and Concept
1993-16 * M. Jarke, K. Pohl, St. Jacobs et al.: Requirements Engineering: An Integrated View of Representation Process and Domain
1993-17 * M. Jarke, K. Pohl: Establishing Vision in Context: Towards a Model of Requirements Processes
1993-18 W. Hans, H. Kuchen, St. Winkler: Full Indexing for Lazy Narrowing
1993-19 W. Hans, J.J. Ruz, F. Saenz, St. Winkler: A VHDL Specification of a Shared Memory Parallel Machine for Babel
1993-20 * K. Finke, M. Jarke, P. Szczurko, R. Soltysiak: Quality Management for Expert Systems in Process Control
1993-21 M. Jarke, M.A. Jeusfeld, P. Szczurko: Three Aspects of Intelligent Cooperation in the Quality Cycle
1994-01 Margit Generet, Sven Martin (eds.), Fachgruppe Informatik: Jahresbericht 1993
1994-02 M. Lefering: Development of Incremental Integration Tools Using Formal Specifications
1994-03 * P. Constantopoulos, M. Jarke, J. Mylopoulos, Y. Vassiliou: The Software Information Base: A Server for Reuse
1994-04 * Rolf Hager, Rudolf Mathar, Jürgen Mattfeldt: Intelligent Cruise Control and Reliable Communication of Mobile Stations
1994-05 * Rolf Hager, Peter Hermesmann, Michael Portz: Feasibility of Authentication Procedures within Advanced Transport Telematics
1994-06 * Claudia Popien, Bernd Meyer, Axel Kuepper: A Formal Approach to Service Import in ODP Trader Federations
1994-07 P. Peters, P. Szczurko: Integrating Models of Quality Management Methods by an Object-Oriented Repository
1994-08 * Manfred Nagl, Bernhard Westfechtel: A Universal Component for the Administration in Distributed and Integrated Development Environments
1994-09 * Patrick Horster, Holger Petersen: Signatur- und Authentifikationsverfahren auf der Basis des diskreten Logarithmusproblems
1994-11 A. Schürr: PROGRES, A Visual Language and Environment for PROgramming with Graph REwrite Systems
1994-12 A. Schürr: Specification of Graph Translators with Triple Graph Grammars
1994-13 A. Schürr: Logic Based Programmed Structure Rewriting Systems
1994-14 L. Staiger: Codes, Simplifying Words, and Open Set Condition
1994-15 * Bernhard Westfechtel: A Graph-Based System for Managing Configurations of Engineering Design Documents
1994-16 P. Klein: Designing Software with Modula-3
1994-17 I. Litovsky, L. Staiger: Finite acceptance of infinite words

1994-18 G. Hogen, R. Loogen: Parallel Functional Implementations: Graphbased vs. Stackbased Reduction

1994-19 M. Jeusfeld, U. Johnen: An Executable Meta Model for Re-Engineering of Database Schemas

1994-20 * R. Gallersdörfer, M. Jarke, K. Klabunde: Intelligent Networks as a Data Intensive Application (INDIA)

1994-21 M. Mohnen: Proving the Correctness of the Static Link Technique Using Evolving Algebras

1994-22 H. Fernau, L. Staiger: Valuations and Unambiguity of Languages, with Applications to Fractal Geometry

1994-24 * M. Jarke, K. Pohl, R. Dömges, St. Jacobs, H. W. Nissen: Requirements Information Management: The NATURE Approach

1994-25 * M. Jarke, K. Pohl, C. Rolland, J.-R. Schmitt: Experience-Based Method Evaluation and Improvement: A Process Modeling Approach

1994-26 * St. Jacobs, St. Kethers: Improving Communication and Decision Making within Quality Function Deployment

1994-27 * M. Jarke, H. W. Nissen, K. Pohl: Tool Integration in Evolving Information Systems Environments

1994-28 O. Burkart, D. Caucal, B. Steffen: An Elementary Bisimulation Decision Procedure for Arbitrary Context-Free Processes

1995-01 * Fachgruppe Informatik: Jahresbericht 1994

1995-02 Andy Schürr, Andreas J. Winter, Albert Zündorf: Graph Grammar Engineering with PROGRES

1995-03 Ludwig Staiger: A Tight Upper Bound on Kolmogorov Complexity by Hausdorff Dimension and Uniformly Optimal Prediction

1995-04 Birgitta König-Ries, Sven Helmer, Guido Moerkotte: An experimental study on the complexity of left-deep join ordering problems for cyclic queries

1995-05 Sophie Cluet, Guido Moerkotte: Efficient Evaluation of Aggregates on Bulk Types

1995-06 Sophie Cluet, Guido Moerkotte: Nested Queries in Object Bases

1995-07 Sophie Cluet, Guido Moerkotte: Query Optimization Techniques Exploiting Class Hierarchies

1995-08 Markus Mohnen: Efficient Compile-Time Garbage Collection for Arbitrary Data Structures

1995-09 Markus Mohnen: Functional Specification of Imperative Programs: An Alternative Point of View of Functional Languages

1995-10 Rainer Gallersdörfer, Matthias Nicola: Improving Performance in Replicated Databases through Relaxed Coherency

1995-11 * M.Staudt, K.von Thadden: Subsumption Checking in Knowledge Bases

1995-12 * G.V.Zemanek, H.W.Nissen, H.Hubert, M.Jarke: Requirements Analysis from Multiple Perspectives: Experiences with Conceptual Modeling Technology

1995-13 * M.Staudt, M.Jarke: Incremental Maintenance of Externally Materialized Views

1995-14 * P.Peters, P.Szczurko, M.Jeusfeld: Oriented Information Management: Conceptual Models at Work

1995-15 * Matthias Jarke, Sudha Ram (Hrsg.): WITS 95 Proceedings of the 5th Annual Workshop on Information Technologies and Systems

1995-16 * W.Hans, St.Winkler, F.Saenz: Distributed Execution in Functional Logic Programming

1996-01 * Jahresbericht 1995

1996-02 Michael Hanus, Christian Prehofer: Higher-Order Narrowing with Definitional Trees

1996-03 * W.Scheufele, G.Moerkotte: Optimal Ordering of Selections and Joins in Acyclic Queries with Expensive Predicates

1996-04 Klaus Pohl: PRO-ART: Enabling Requirements Pre-Traceability

1996-05 Klaus Pohl: Requirements Engineering: An Overview

1996-06 * M.Jarke, W.Marquardt: Design and Evaluation of Computer–Aided Process Modelling Tools

1996-07 Olaf Chitil: The Sigma-Semantics: A Comprehensive Semantics for Functional Programs

1996-08 * S.Sripada: On Entropy and the Limitations of the Second Law of Thermodynamics

1996-09 Michael Hanus (Ed.): Proceedings of the Poster Session of ALP96 - Fifth International Conference on Algebraic and Logic Programming

1996-09-0 Michael Hanus (Ed.): Proceedings of the Poster Session of ALP 96 - Fifth International Conference on Algebraic and Logic Programming: Introduction and table of contents

1996-09-1 Ilies Alouini: An Implementation of Conditional Concurrent Rewriting on Distributed Memory Machines

1996-09-2 Olivier Danvy, Karoline Malmkjær: On the Idempotence of the CPS Transformation

1996-09-3 Víctor M. Gulías, José L. Freire: Concurrent Programming in Haskell

1996-09-4 Sébastien Limet, Pierre Réty: On Decidability of Unifiability Modulo Rewrite Systems

1996-09-5 Alexandre Tessier: Declarative Debugging in Constraint Logic Programming

1996-10 Reidar Conradi, Bernhard Westfechtel: Version Models for Software Configuration Management

1996-11 * C.Weise, D.Lenzkes: A Fast Decision Algorithm for Timed Refinement

1996-12 * R.Dömges, K.Pohl, M.Jarke, B.Lohmann, W.Marquardt: PRO-ART/CE* — An Environment for Managing the Evolution of Chemical Process Simulation Models

1996-13 * K.Pohl, R.Klamma, K.Weidenhaupt, R.Dömges, P.Haumer, M.Jarke: A Framework for Process-Integrated Tools

1996-14 * R.Gallersdörfer, K.Klabunde, A.Stolz, M.Eßmajor: INDIA — Intelligent Networks as a Data Intensive Application, Final Project Report, June 1996

1996-15 * H.Schimpe, M.Staudt: VAREX: An Environment for Validating and Refining Rule Bases

1996-16 * M.Jarke, M.Gebhardt, S.Jacobs, H.Nissen: Conflict Analysis Across Heterogeneous Viewpoints: Formalization and Visualization

1996-17 Manfred A. Jeusfeld, Tung X. Bui: Decision Support Components on the Internet

1996-18 Manfred A. Jeusfeld, Mike Papazoglou: Information Brokering: Design, Search and Transformation

1996-19 * P.Peters, M.Jarke: Simulating the impact of information flows in networked organizations

1996-20 Matthias Jarke, Peter Peters, Manfred A. Jeusfeld: Model-driven planning and design of cooperative information systems

1996-21 * G.de Michelis, E.Dubois, M.Jarke, F.Matthes, J.Mylopoulos, K.Pohl, J.Schmidt, C.Woo, E.Yu: Cooperative information systems: a manifesto

1996-22 * S.Jacobs, M.Gebhardt, S.Kethers, W.Rzasa: Filling HTML forms simultaneously: CoWeb architecture and functionality

1996-23 * M.Gebhardt, S.Jacobs: Conflict Management in Design

1997-01 Michael Hanus, Frank Zartmann (eds.): Jahresbericht 1996

1997-02 Johannes Faassen: Using full parallel Boltzmann Machines for Optimization

1997-03 Andreas Winter, Andy Schürr: Modules and Updatable Graph Views for PROgrammed Graph REwriting Systems

1997-04 Markus Mohnen, Stefan Tobies: Implementing Context Patterns in the Glasgow Haskell Compiler

1997-05 * S.Gruner: Schemakorrespondenzaxiome unterstützen die paargrammatische Spezifikation inkrementeller Integrationswerkzeuge

1997-06 Matthias Nicola, Matthias Jarke: Design and Evaluation of Wireless Health Care Information Systems in Developing Countries

1997-07 Petra Hofstedt: Taskparallele Skelette für irregulär strukturierte Probleme in deklarativen Sprachen

1997-08 Dorothea Blostein, Andy Schürr: Computing with Graphs and Graph Rewriting

1997-09 Carl-Arndt Krapp, Bernhard Westfechtel: Feedback Handling in Dynamic Task Nets

1997-10 Matthias Nicola, Matthias Jarke: Integrating Replication and Communication in Performance Models of Distributed Databases

1997-11 * R. Klamma, P. Peters, M. Jarke: Workflow Support for Failure Management in Federated Organizations

1997-13 Markus Mohnen: Optimising the Memory Management of Higher-Order Functional Programs

1997-14 Roland Baumann: Client/Server Distribution in a Structure-Oriented Database Management System

1997-15 George Botorog: High-Level Parallel Programming and the Efficient Implementation of Numerical Algorithms

1998-01 * Fachgruppe Informatik: Jahresbericht 1997

1998-02 Stefan Gruner, Manfred Nagel, Andy Schürr: Fine-grained and Structure-Oriented Document Integration Tools are Needed for Development Processes

1998-03 Stefan Gruner: Einige Anmerkungen zur graphgrammatischen Spezifikation von Integrationswerkzeugen nach Westfechtel, Janning, Lefering und Schürr

1998-04 * O. Kubitz: Mobile Robots in Dynamic Environments

1998-05 Martin Leucker, Stephan Tobies: Truth - A Verification Platform for Distributed Systems

1998-06 * Matthias Oliver Berger: DECT in the Factory of the Future
1998-07 M. Arnold, M. Erdmann, M. Glinz, P. Haumer, R. Knoll, B. Paech, K. Pohl, J. Ryser, R. Studer, K. Weidenhaupt: Survey on the Scenario Use in Twelve Selected Industrial Projects
1998-08 * H. Aust: Sprachverstehen und Dialogmodellierung in natürlichsprachlichen Informationssystemen
1998-09 * Th. Lehmann: Geometrische Ausrichtung medizinischer Bilder am Beispiel intraoraler Radiographien
1998-10 * M. Nicola, M. Jarke: Performance Modeling of Distributed and Replicated Databases
1998-11 * Ansgar Schleicher, Bernhard Westfechtel, Dirk Jäger: Modeling Dynamic Software Processes in UML
1998-12 * W. Appelt, M. Jarke: Interoperable Tools for Cooperation Support using the World Wide Web
1998-13 Klaus Indermark: Semantik rekursiver Funktionsdefinitionen mit Striktheitsinformation
1999-01 * Jahresbericht 1998
1999-02 * F. Huch: Verifcation of Erlang Programs using Abstract Interpretation and Model Checking — Extended Version
1999-03 * R. Gallersdörfer, M. Jarke, M. Nicola: The ADR Replication Manager
1999-04 María Alpuente, Michael Hanus, Salvador Lucas, Germán Vidal: Specialization of Functional Logic Programs Based on Needed Narrowing
1999-05 * W. Thomas (Ed.): DLT 99 - Developments in Language Theory Fourth International Conference
1999-06 * Kai Jakobs, Klaus-Dieter Kleefeld: Informationssysteme für die angewandte historische Geographie
1999-07 Thomas Wilke: CTL+ is exponentially more succinct than CTL
1999-08 Oliver Matz: Dot-Depth and Monadic Quantifier Alternation over Pictures
2000-01 * Jahresbericht 1999
2000-02 Jens Vöge, Marcin Jurdzinski: A Discrete Strategy Improvement Algorithm for Solving Parity Games
2000-04 Andreas Becks, Stefan Sklorz, Matthias Jarke: Exploring the Semantic Structure of Technical Document Collections: A Cooperative Systems Approach
2000-05 Mareike Schoop: Cooperative Document Management
2000-06 Mareike Schoop, Christoph Quix (eds.): Proceedings of the Fifth International Workshop on the Language-Action Perspective on Communication Modelling
2000-07 * Markus Mohnen, Pieter Koopman (Eds.): Proceedings of the 12th International Workshop of Functional Languages
2000-08 Thomas Arts, Thomas Noll: Verifying Generic Erlang Client-Server Implementations
2001-01 * Jahresbericht 2000
2001-02 Benedikt Bollig, Martin Leucker: Deciding LTL over Mazurkiewicz Traces
2001-03 Thierry Cachat: The power of one-letter rational languages

2001-04 Benedikt Bollig, Martin Leucker, Michael Weber: Local Parallel Model Checking for the Alternation Free mu-Calculus
2001-05 Benedikt Bollig, Martin Leucker, Thomas Noll: Regular MSC Languages
2001-06 Achim Blumensath: Prefix-Recognisable Graphs and Monadic Second-Order Logic
2001-07 Martin Grohe, Stefan Wöhrle: An Existential Locality Theorem
2001-08 Mareike Schoop, James Taylor (eds.): Proceedings of the Sixth International Workshop on the Language-Action Perspective on Communication Modelling
2001-09 Thomas Arts, Jürgen Giesl: A collection of examples for termination of term rewriting using dependency pairs
2001-10 Achim Blumensath: Axiomatising Tree-interpretable Structures
2001-11 Klaus Indermark, Thomas Noll (eds.): Kolloquium Programmiersprachen und Grundlagen der Programmierung
2002-01 * Jahresbericht 2001
2002-02 Jürgen Giesl, Aart Middeldorp: Transformation Techniques for Context-Sensitive Rewrite Systems
2002-03 Benedikt Bollig, Martin Leucker, Thomas Noll: Generalised Regular MSC Languages
2002-04 Jürgen Giesl, Aart Middeldorp: Innermost Termination of Context-Sensitive Rewriting
2002-05 Horst Lichter, Thomas von der Maßen, Thomas Weiler: Modelling Requirements and Architectures for Software Product Lines
2002-06 Henry N. Adorna: 3-Party Message Complexity is Better than 2-Party Ones for Proving Lower Bounds on the Size of Minimal Nondeterministic Finite Automata
2002-07 Jörg Dahmen: Invariant Image Object Recognition using Gaussian Mixture Densities
2002-08 Markus Mohnen: An Open Framework for Data-Flow Analysis in Java
2002-09 Markus Mohnen: Interfaces with Default Implementations in Java
2002-10 Martin Leucker: Logics for Mazurkiewicz traces
2002-11 Jürgen Giesl, Hans Zantema: Liveness in Rewriting
2003-01 * Jahresbericht 2002
2003-02 Jürgen Giesl, René Thiemann: Size-Change Termination for Term Rewriting
2003-03 Jürgen Giesl, Deepak Kapur: Deciding Inductive Validity of Equations
2003-04 Jürgen Giesl, René Thiemann, Peter Schneider-Kamp, Stephan Falke: Improving Dependency Pairs
2003-05 Christof Löding, Philipp Rohde: Solving the Sabotage Game is PSPACE-hard
2003-06 Franz Josef Och: Statistical Machine Translation: From Single-Word Models to Alignment Templates
2003-07 Horst Lichter, Thomas von der Maßen, Alexander Nyßen, Thomas Weiler: Vergleich von Ansätzen zur Feature Modellierung bei der Softwareproduktlinienentwicklung
2003-08 Jürgen Giesl, René Thiemann, Peter Schneider-Kamp, Stephan Falke: Mechanizing Dependency Pairs
2004-01 * Fachgruppe Informatik: Jahresbericht 2004

2004-02	Benedikt Bollig, Martin Leucker: Message-Passing Automata are expressively equivalent to EMSO logic
2004-03	Delia Kesner, Femke van Raamsdonk, Joe Wells (eds.): HOR 2004 – 2nd International Workshop on Higher-Order Rewriting
2004-04	Slim Abdennadher, Christophe Ringeissen (eds.): RULE 04 – Fifth International Workshop on Rule-Based Programming
2004-05	Herbert Kuchen (ed.): WFLP 04 – 13th International Workshop on Functional and (Constraint) Logic Programming
2004-06	Sergio Antoy, Yoshihito Toyama (eds.): WRS 04 – 4th International Workshop on Reduction Strategies in Rewriting and Programming
2004-07	Michael Codish, Aart Middeldorp (eds.): WST 04 – 7th International Workshop on Termination
2004-08	Klaus Indermark, Thomas Noll: Algebraic Correctness Proofs for Compiling Recursive Function Definitions with Strictness Information
2004-09	Joachim Kneis, Daniel Mölle, Stefan Richter, Peter Rossmanith: Parameterized Power Domination Complexity
2004-10	Zinaida Benenson, Felix C. Gärtner, Dogan Kesdogan: Secure Multi-Party Computation with Security Modules
2005-01 *	Fachgruppe Informatik: Jahresbericht 2005
2005-02	Maximillian Dornseif, Felix C. Gärtner, Thorsten Holz, Martin Mink: An Offensive Approach to Teaching Information Security: „Aachen Summer School Applied IT Security“
2005-03	Jürgen Giesl, René Thiemann, Peter Schneider-Kamp: Proving and Disproving Termination of Higher-Order Functions
2005-04	Daniel Mölle, Stefan Richter, Peter Rossmanith: A Faster Algorithm for the Steiner Tree Problem

* These reports are only available as a printed version.
Please contact `biblio@informatik.rwth-aachen.de` to obtain copies.