



Volume 2, Issue 2, February 2012

Network Attack Detection and Defense Early Warning Systems – Challenges and Perspectives (Dagstuhl Seminar 12061) <i>Georg Carle, Hervé Debar, Falko Dressler, and Hartmut König</i>	1
Software Clone Management Towards Industrial Application (Dagstuhl Seminar 12071) <i>Rainer Koschke, Ira D. Baxter, Michael Conradt, and James R. Cordy</i>	21
Information Visualization, Visual Data Mining and Machine Learning (Dagstuhl Seminar 12081) <i>Daniel A. Keim, Fabrice Rossi, Thomas Seidl, Michel Verleysen, and Stefan Wrobel</i>	58
Principles of Provenance (Dagstuhl Seminar 12091) <i>James Cheney, Anthony Finkelstein, Bertram Ludäscher, and Stijn Vansummeren</i>	84

ISSN 2192-5283

Published online and open access by

Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany.

Online available at <http://www.dagstuhl.de/dagrep>

Publication date

June, 2012

Bibliographic information published by the Deutsche Nationalbibliothek

The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at <http://dnb.d-nb.de>.

License

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported license: CC-BY-NC-ND.



In brief, this license authorizes each and everybody to share (to copy, distribute and transmit) the work under the following conditions, without impairing or restricting the authors' moral rights:

- Attribution: The work must be attributed to its authors.
- Noncommercial: The work may not be used for commercial purposes.
- No derivation: It is not allowed to alter or transform this work.

The copyright is retained by the corresponding authors.

Aims and Scope

The periodical *Dagstuhl Reports* documents the program and the results of Dagstuhl Seminars and Dagstuhl Perspectives Workshops.

In principal, for each Dagstuhl Seminar or Dagstuhl Perspectives Workshop a report is published that contains the following:

- an executive summary of the seminar program and the fundamental results,
 - an overview of the talks given during the seminar (summarized as talk abstracts), and
 - summaries from working groups (if applicable).
- This basic framework can be extended by suitable contributions that are related to the program of the seminar, e.g. summaries from panel discussions or open problem sessions.

Editorial Board

- Susanne Albers
- Bernd Becker
- Karsten Berns
- Stephan Diehl
- Hannes Hartenstein
- Frank Leymann
- Stephan Merz
- Bernhard Nebel
- Han La Poutré
- Bernt Schiele
- Nicole Schweikardt
- Raimund Seidel
- Gerhard Weikum
- Reinhard Wilhelm (*Editor-in-Chief*)

Editorial Office

Marc Herbstritt (*Managing Editor*)
Jutka Gasirowski (*Editorial Assistance*)
Thomas Schillo (*Technical Assistance*)

Contact

Schloss Dagstuhl – Leibniz-Zentrum für Informatik
Dagstuhl Reports, Editorial Office
Oktavie-Allee, 66687 Wadern, Germany
reports@dagstuhl.de

Digital Object Identifier: 10.4230/DagRep.2.2.i

www.dagstuhl.de/dagrep

Network Attack Detection and Defense Early Warning Systems – Challenges and Perspectives

Edited by

Georg Carle¹, Hervé Debar², Falko Dressler³, and Hartmut König⁴

1 TU München, DE, carle@in.tum.de

2 Télécom SudParis, Evry, FR, herve.debar@telecom-sudparis.eu

3 Universität Innsbruck, AT, falko.dressler@uibk.ac.at on behalf of Jelena Mirkovic

4 BTU Cottbus, DE, koenig@informatik.tu-cottbus.de

Abstract

The increasing dependence of human society on information technology (IT) systems requires appropriate measures to cope with their misuse. The growing potential of threats, which make these systems more and more vulnerable, is caused by the complexity of the technologies themselves. The potential of threats in networked systems will further grow as well as the number of individuals who are able to abuse these systems. It becomes increasingly apparent that IT security cannot be achieved by prevention alone. Preventive measures and reactive aspects need to complement one another. A major challenge of modern IT security technologies is to cope with an exploding variability of attacks which stems from a significant commercial motivation behind them. Increasingly proactive measures are required to ward off these threats.

Increased efforts in research and society are required to protect critical civil infrastructures, such as the health care system, the traffic system, power supply, trade, military networks, and others in developed countries. This is a consequence of the increasing shift of industrial IT systems to the IP protocol leading to sensible IT infrastructures which are more vulnerable as the proprietary systems used in the past. The abundance of services of modern infrastructures critically depends on information and communication technologies. Though, being key enablers of critical infrastructures, these technologies are, at the same time, reckoned among the most vulnerable elements of the whole system. The cooperative information exchange between institutions is mandatory in order to detect distributed and coordinated attacks. Based on a large-scale acquisition of pertinent information, *Early Warning Systems* are a currently pursued approach to draw up situation pictures that allows the detection of trends and upcoming threats, allowing furthermore taking appropriate measures.

The Dagstuhl seminar brought together researchers from academia and industry. The objective of the seminar was to further discuss challenges and methods in the area of attack detection and defense. The seminar was supposed to focus on design aspects of early warning systems and related monitoring infrastructures, e.g., intrusion detection overlays, to protect computer systems, networks, and critical infrastructures. The seminar was jointly organized by Georg Carle, Hervé Debar, Hartmut König, and Jelena Mirkovic. It was attended by 34 participants from nine countries.

Seminar 05.–10. February, 2012 – www.dagstuhl.de/12061

1998 ACM Subject Classification K.6.5 Security and Protection, K.4.2 Social Issues

Keywords and phrases early warning systems, critical infrastructure protection, botnets, intrusion detection, malware assessment, vulnerability analysis, network monitoring, flow analysis, denial-of-service detection and response, event correlation, attack response and countermeasures

Digital Object Identifier 10.4230/DagRep.2.2.1

Edited in cooperation with Franka Schuster (BTU Cottbus, DE)



Except where otherwise noted, content of this report is licensed under a Creative Commons BY-NC-ND 3.0 Unported license

Network Attack Detection and Defense Early Warning Systems, *Dagstuhl Reports*, Vol. 2, Issue 2, pp. 1–20

Editors: Georg Carle, Hervé Debar, Falko Dressler, and Hartmut König




Dagstuhl Reports

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Executive Summary

Georg Carle
 Hervé Debar
 Falko Dressler
 Hartmut König

License  Creative Commons BY-NC-ND 3.0 Unported license
 © Georg Carle, Hervé Debar, Falko Dressler, and Hartmut König

The objective of the seminar was to discuss new challenges, technologies, and architectures in the area of network attack detection and defense. The focus of this seminar laid in particular on *early warning systems*, *malware detection*, and the *protection of critical infrastructures*, but also other recently emerging topics were supposed to be discussed. On this account, the seminar consisted of plenary sessions with technical talks and various breakout sessions. Beside the topics mentioned above two other topics on recently emerging issues were added, namely *cyber crime versus cyber war* and the *protection of cyber-physical systems*.

The seminar started off with an introductory session in which all participants shortly introduced themselves and discussed the focus and the structure of the seminar. Thereafter the first topic *Challenges on Early Warning Systems and Malware Detection* was raised. Michael Meier gave a state of the art talk on the development of early warning systems in the last years and open issues. Felix C. Freiling and Falko Dressler reported on the results of their projects in this field with the German Federal Office for Information Security (BSI). Jan Kohlrausch gave an overview of the experience with the deployment of early warning systems in practice with the DFN-CERT. In the afternoon the first breakout sessions were held. The topics discussed were the *Future of Early Warning Systems*, *Cloud Security*, and *Teaching IT Security*.

Tuesday was devoted to the topic *Protection of Critical Infrastructures*. Introductory talks of the various aspects and challenges for protecting critical infrastructures were given by Stephen Wolthusen and Corrado Leita, followed by technical talks by Franka Schuster and Andreas Paul about a project for protecting supervisory control and data acquisition (SCADA) networks, by Simin Nadjm-Tehrani on the security of smart meters, and by Georg Carle, Lothar Braun and Holger Kinkelin on large-scale vulnerability assessment. In the afternoon Jens Tölle spoke about the protection of IP infrastructures with model-based cyber defense situational awareness. After coffee break we continued with two further breakout sessions on *Information Security for Novel Devices* and *Fighting against Botnets*.

Wednesday morning was devoted to two special topics which have emerged recently: *Security of Cyber-Physical Systems* and *Cyber Crime versus Cyber War*. Nils Aschenbruck gave an introductory talk to the first topic reflecting the evolution from sensor networks to cyber-physical systems. Falko Dressler addressed in his talk the security challenges for future nano communication. The discussion on this topic was continued in the breakout session on Thursday. The second topic was opened by Felix C. Freiling posing various questions about the differences between malware for the masses and exclusive malware, and how to detect them as basis for a longer discussion in the auditorium. Gabi Dreö Rodosek then elucidated at length the issue in her talk about cyber defense. In the afternoon we made a nice trip to the historic city of Trier. The pretty cold weather there gave many opportunities to continue the discussions in warm coffee shops.

On Thursday morning we commenced with two talks by Pavel Laskov and Konrad Rieck on *Malware Detection* which dealt especially with machine learning aspects. Sven Dietrich added a talk on his SkyNET project about the use of drones to launch attacks on wireless

networks. Thereafter we continued the topic on the protection of critical infrastructures with the focus on new challenges in deep packet inspection. Radu State began with a talk on the semantic exploration of DNS domains. René Rietz continued with a talk on the increasing threat by attacks over the web. After lunch Robin Sommer introduced the new version of the intrusion detection system (IDS) Bro. Alexander von Gernler reported about the current practice of application level firewalling and virus scanning from the perspective of a firewall manufacturer. Finally, Michael Vogel presented an approach for a dynamically adapting multi-agent intrusion detection system which copes with the growing gap between the evolution of network bandwidth and the single-thread performance of today's CPU architectures. After the coffee break, two further breakout sessions on cyber-physical systems and smart energy grids took place.

Friday morning hosted two talks by Bettina Schnor and Simin Nadjm-Tehrani on IPv6 security and anomaly detection in mobile networks. After that we concluded the seminar with a discussion about the seminar outcome and possible future seminars.

Conclusion

The seminar was well-received by all participants. It gave a good opportunity to inform about current challenges in the area of network attack detection and defense and discuss possible countermeasures. Especially the breakout sessions found a great acceptance. The participants further liked much the possibility to have detailed discussions with colleagues outside the official program. They regret that not all invited foreign scientist accepted the invitation. They will advertise more strongly for this seminar. All participants agreed that proposal for another seminar should be submitted. There are two concrete contributions of this seminar:

1. Current research results of eight participating groups were published in special issue of the journal PIK 1/2012 which is especially devoted to this Dagstuhl seminar.
2. The discussion during the breakout session on cyber-physical systems showed that there is still an unclear picture on the security challenges to these systems. This raised the idea to apply for a Dagstuhl perspective workshop to discuss in detail the security challenges for protecting cyber-physical systems and to define them in a manifesto as working base for further research activities. The proposal has been submitted meanwhile.

2 Table of Contents

Executive Summary

<i>Georg Carle, Hervé Debar, Falko Dressler, and Hartmut König</i>	2
--	---

Overview of Talks


Early Warning Systems <i>Michael Meier</i>	6
Early Warning Systems – a German Project Initiative <i>Falko Dressler</i>	6
Early Warning Systems: Experiences from InMAS <i>Felix C. Freiling</i>	6
Early Warning and Malware Detection at the DFN-CERT <i>Jan Kohlrausch</i>	7
Critical Infrastructure Protection <i>Stephen Wolthusen</i>	7
Challenges in Critical Infrastructure Security <i>Corrado Leita</i>	7
Protecting Critical Infrastructures <i>Franka Schuster and Andreas Paul</i>	8
Large-scale Vulnerability Assessment Using Active and Passive Techniques <i>Georg Carle, Lothar Braun, and Holger Kinkel</i>	8
Protecting IP Infrastructures with Model-based Cyber Defense Situational Awareness <i>Jens Tölle</i>	9
SecFutur: Security Engineering Process for Networked Embedded Devices <i>Simin Nadjm-Tehrani</i>	9
From WSN to CPS Security – How Crucial are the Remaining Challenges <i>Nils Aschenbruck</i>	10
Going Nano – A New Playground and Novel Challenges for Security <i>Falko Dressler</i>	10
Attack Detection 2.0: Detecting High-Quality Attacks <i>Felix C. Freiling</i>	10
Cyber Defense: A View from the Research Perspective <i>Gabi Dreo Rodosek</i>	11
The Threat of Love Letters: Detection of Document-based Attacks <i>Pavel Laskov</i>	12
Learning-based Defenses against Malicious JavaScript Code <i>Konrad Rieck</i>	12
Semantic Exploration of DNS <i>Radu State</i>	12
Intrusion Detection for the Web 2.0 <i>René Rietz</i>	13

From the Unexpected Side: SkyNET	
<i>Sven Dietrich</i>	13
State of the Art and Limitations to Application Level Firewalling and Virus Scanning	
<i>Alexander von Gernler</i>	13
Bro 2.0 and Beyond	
<i>Robin Sommer</i>	14
A Dynamically Adapting Distributed Multi-Agent IDS	
<i>Michael Vogel</i>	14
Security Challenges of IPv6 Networks	
<i>Bettina Schnor</i>	15
Anomaly Detection in Challenged Networks	
<i>Simin Nadjm-Tehrani</i>	15
Working Groups	
Breakout Session: Future of Early Warning Systems	
<i>Bettina Schnor</i>	16
Breakout Session: Cloud Security	
<i>Holger Kinkel</i>	16
Breakout Session: Teaching IT Security	
<i>Hervé Debar</i>	17
Breakout Session: Information Security for Novel Devices	
<i>Elmar Gerhards-Padilla</i>	17
Breakout Session: Fighting against Botnets	
<i>Michael Meier</i>	18
Breakout Session: Smart Energy Grids	
<i>Michael Vogel</i>	18
Breakout Session: Critical Infrastructure Protection	
<i>Franka Schuster</i>	18
Breakout Session: Cyber-Physical System Security	
<i>Hartmut König</i>	19
Participants	20

3 Overview of Talks

3.1 Early Warning Systems

Michael Meier (Universität Dortmund, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Michael Meier

The talk presents a definition of early warning systems and sketches a number of research projects on early warning systems, namely InMAS (Internet Malware Analysis System), IAS (Internet Analysis System), and AMSEL, as well as the operational early warning system CarmentiS and the Deutsche Telekom early warning system. Further, the different meanings of the term “early” – incomplete and fast are discussed, and the question how fast early warning systems should be able to operate is addressed. The talk concludes with some open questions in the context of early warning systems.

3.2 Early Warning Systems – a German Project Initiative

Falko Dressler (Universität Innsbruck, AT)


License  Creative Commons BY-NC-ND 3.0 Unported license
© Falko Dressler

This talk briefly reflects the requirements and challenges on early warning systems from a technical perspective focusing on the lower layers, i.e., the network sensors and high speed monitoring systems. The crucial issue is to collect network data at highest speeds and to process it in a distributed manner – even given unlimited processing power, it would not be possible to send data for later analysis to any central point in the network.

In the scope of the monkit project, methods for multi-core support and flow analysis with aggregated payload information have been developed. Tools, such as DPA (dialog based payload aggregation), have been implemented in the Vermont monitoring framework and show extremely satisfying results. Still, many issues remain open such as privacy aware data collection and algorithmic solutions to 10 Gbit/s monitoring supporting the use of multiple IDS in presence of correlated flows.

3.3 Early Warning Systems: Experiences from InMAS

Felix C. Freiling (Universität Erlangen-Nürnberg, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Felix C. Freiling

InMAS is a large-scale sensor system for malware built within a project at Universität Mannheim several years ago [1]. The talk is a report on experiences with this project, especially some results of a large-scale data analysis of autonomously spreading malware [2].


References

- 1 Markus Engelberth, Felix C. Freiling, Jan Göbel, Christian Gorecki, Thorsten Holz, Ralf Hund, Philipp Trinius, Carsten Willems. *The InMAS Approach*. 1st European Workshop on Internet Early Warning and Network Intelligence (EWNI), Hamburg, Germany, 2010

- 2 Jan Göbel, Philipp Trinius. *Towards Optimal Sensor Placement Strategies for Early Warning Systems*. Proceedings 5th Conference Sicherheit, Schutz und Zuverlässigkeit (SICHERHEIT), Berlin, Germany, 2010

3.4 Early Warning and Malware Detection at the DFN-CERT


Jan Kohlrausch (DFN-CERT Services GmbH, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Jan Kohlrausch

Aim of this talk is to give an insight into the work and experiences of the DFN-CERT that are related to early warning systems (EWS) and malware detection. First, EWS, such as the CarmentiS system, which was funded by the German BSI, allow one to assess the overall threat level of the German research network (DFN) as well as the Internet. This information is vital to react to global threats, such as large-scale DDoS attacks and new Internet worms. Apart from this, CarmentiS provides data of compromised computer systems which are used for the automatic warning service of the DFN-CERT. This service collects data about compromised systems from different sources and distributes them automatically to the affected sites within the DFN. Other data sources are the honeypots Dionaea and Argos. In addition, Dionaea is designed to capture malware to be analyzed in a sandbox environment. However, the malware constantly improves and may be able to evade the analysis in the future. Furthermore, current IDS and honeypots have to be adapted to cope with IPv6 which grows in importance.

3.5 Critical Infrastructure Protection


Stephen Wolthusen (RHUL – London, GB)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Stephen Wolthusen

Starting from a definition of critical infrastructures and their relation to network attack detection and defense the talk presents various models applied for critical infrastructures. It describes existing dependencies and interdependencies in critical infrastructures using qualitative and quantitative models. In the second part of the talk the attacker model for critical infrastructures is discussed by referring to new challenges, especially in the field of SCADA systems and smart grids. Finally, the objectives of the EU ARTEMIS project are outlined.

3.6 Challenges in Critical Infrastructure Security

Corrado Leita (Symantec Research Labs – Sophia Antipolis, FR)


License  Creative Commons BY-NC-ND 3.0 Unported license
© Corrado Leita

This presentation provides a high-level overview of the security challenges associated with the protection of critical infrastructure environments. Thanks to the convergence between

standard IT systems and industrial control systems (ICS), a set of new challenges and opportunities can be identified when trying to secure these environments. How far can standard IT security techniques go in protecting critical infrastructure environments? Are there special constraints and operational characteristics that are unique to ICS and that render the current state of the art impossible to adapt? The talk tries to walk the audience through the implications associated with these questions.

3.7 Protecting Critical Infrastructures

Franka Schuster and Andreas Paul (BTU Cottbus, DE)


License  Creative Commons BY-NC-ND 3.0 Unported license
© Franka Schuster and Andreas Paul

Joint work of Schuster, Franka; Paul, Andreas; Vogel, Michael; Rietz, René

The deployment of common information and communication technology in SCADA systems and the recent use of Industrial Ethernet (IE) down to the field level induce new security risks to critical infrastructures. In the talk the drawbacks of current security measures in critical infrastructures are discussed to address the lack of a highly tailored intrusion detection system. First a brief introduction of the concepts and vulnerabilities of Profinet as an example of an emerging IE protocol is given. On this basis, a novel approach for a distributed intrusion detection system is presented which is specialized in the analysis of network traffic of SCADA protocols, such as Profinet.

3.8 Large-scale Vulnerability Assessment Using Active and Passive Techniques

Georg Carle, Lothar Braun, and Holger Kinkel (TU München, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Georg Carle, Lothar Braun, and Holger Kinkel

Joint work of Carle, Georg; Braun, Lothar; Kinkel, Holger


Current intrusion detection systems detect attacks, while they are conducted or search for signs of successful previous attacks. If one considers previous outbreaks of worms or other exploitations of vulnerabilities in computer systems, one can often find that attackers exploit vulnerabilities which have been known for some time. Many exploited vulnerabilities have been fixed by their vendors before the first exploit for this vulnerability has been observed. Administrators of vulnerable systems therefore usually have time to fix vulnerabilities before they are exploited.

Our work aims at finding weaknesses in systems and security infrastructures to provide administrators and users with a security assessment of their deployed infrastructure. Our presentation discusses how Internet-wide large-scale network measurements can be used to assess the current deployment of computer systems and security protocols. Strengths and weaknesses of active and passive measurement techniques are discussed, including limitations that render network measurements ineffective.

Obtaining data in scenarios where network measurements are unfeasible is addressed as well. Hence, the presentation also discusses how to obtain measurement data from host-based sensors, such as host-based intrusion detection systems, in a secure and trustworthy way.

3.9 Protecting IP Infrastructures with Model-based Cyber Defense Situational Awareness


Jens Tölle (Fraunhofer FKIE – Wachtberg, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Jens Tölle

The presentation focuses on the improvement of situational awareness in IP networks. Security information and event management (SIEM) systems deliver a huge amount of status data. The presented approach aims at limiting the amount of information which is presented to a human operator/security officer/manager/user in order to help him/her to gain overview without being overloaded by information. In addition, based on a model and a current state gained through measurements, the system gives the possibility to calculate consequences of reactions without applying them to the operational network.

3.10 SecFutur: Security Engineering Process for Networked Embedded Devices

Simin Nadjm-Tehrani (Linköping University, SE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Simin Nadjm-Tehrani


Ask an engineer in the embedded systems sector about the challenges in product development and chances are that the keywords *size*, *performance*, and *cost* will be included in the answer. Indeed the driving forces in the embedded market have been miniaturisation, faster time to market, and higher performance in the past decade. This equation is subject to a rapid change in the years to come. With more embedded devices perpetually connected via a network we see the emergence of security properties among the basic requirements in product development. This is a radical departure from the earlier state of the “things”, where the devices were naturally protected from security threats by operating in closed and controlled environments. Today’s systems are increasingly adopting open standards; and when it comes to networked devices we see the emergence of IP networks in diverse domains, such as the energy sector, banking, and telecommunications.

This dramatic change together with the increased hostility in the operational environment of networked applications makes security requirements a basic tenet that needs to be realized by additional building blocks (e.g., access control, authentication, intrusion monitoring, and forensics). It is also increasingly evident that these requirements cannot be met through an add-on feature developed at late development stages. Efficient development of secure embedded systems requires an engineering process that brings together existing solutions in hardware and software and can be demonstrated to achieve design goals, such as resource efficiency as well as meeting legal and international requirements.

This talk briefly describes the objectives of a three-year European FP7 project addressing security in future networked environments (SecFutur). The project aims to flexibly integrate security solutions into a framework for development of networked embedded systems. During the talk the embedding of an anomaly detector into a smart metering device as work in progress is presented.

3.11 From WSN to CPS Security – How Crucial are the Remaining Challenges

Nils Aschenbruck (Universität Bonn, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Nils Aschenbruck

Joint work of Aschenbruck, Nils; Bauer, Jan; Bieling, Jakob; Bothe, Alexander; Gerhards-Padilla, Elmar; Schwamborn, Matthias


Main reference N. Aschenbruck, J. Bauer, J. Bieling, A. Bothe, M. Schwamborn, P. Martini, D. Pfisterer, K. Hakim, S. Fischer, C. Buschmann, F. Gehring, C. Wiesebrink, “Wireless Sensor Networks-Labor,” Projektbericht, Abschlussdokumentation, 2011.

URL http://net.cs.uni-bonn.de/fileadmin/ag/martini/projekte/wsnlab/wsnlab_abschlussbericht.pdf

Security in wireless sensor networks (WSN) has been an active research area for several years. In the last years different solutions were evaluated in real-world deployments. This helped to highlight the remaining challenges. Cyber-physical systems (CPS) assume a tight combination and coordination of computational and physical resources. WSN (as well as robotics) are typically seen as essential parts of CPS. Thus, the remaining security challenges in the area of WSN do affect CPS. After some motivation and definitions, the talk surveys selected projects and results in the area of WSN and discusses the impact on CPS.

3.12 Going Nano – A New Playground and Novel Challenges for Security

Falko Dressler (Universität Innsbruck, AT)


License  Creative Commons BY-NC-ND 3.0 Unported license
© Falko Dressler

Nano communication is one of the fastest growing emerging research fields. Experts agree that only the interaction among nano machines allows to address the very complex requirements in the field. Drug delivery and environmental control are only two of the many interesting application domains. Relevant communication concepts have been investigated, such as RF radio communication in the terra hertz band or molecular communication based on transmitter molecules. However, one question has not been considered so far and that is nano communication security.

The objective of the talk is to provide some first insights into the security challenges and to highlight some of the open research challenges in this field. A key observation is that especially for molecular communication existing security and cryptographic solutions might not be applicable.

3.13 Attack Detection 2.0: Detecting High-Quality Attacks

Felix C. Freiling (Universität Erlangen-Nürnberg, DE)



License  Creative Commons BY-NC-ND 3.0 Unported license
© Felix C. Freiling

The appearance of the Stuxnet worm has given rise to a new level in the design of attacks and opened a broad discussion on cyber crime versus cyber war. The talk provides a basis for the discussion on the differences between dedicated and mass attacks in this context. What

are the distinguishing features of high-quality targeted attacks in comparison to low-quality (e.g., randomly scanning) malware infections? How can we detect targeted attacks? What research path should we take regarding this? The talk intends to simply pose these questions illustrating them with examples from different categories.

3.14 Cyber Defense: A View from the Research Perspective

Gabi Dreo Rodosek (Universität der Bundeswehr – München, DE)

License     Creative Commons BY-NC-ND 3.0 Unported license
© Gabi Dreo Rodosek

Cyber defense, the defense in the virtual world, refers to countermeasures against IT threats. The communication possibilities are various, from e-mails, the use of P2P applications, such as Skype to Voice over IP, or social networks like Facebook. In addition, the mobility, heterogeneity, the huge number of ubiquitous devices, and the encryption of the data are further challenges to face.

Terms like *cyber war*, *cyber defense*, *cyber threats*, *cyber crime*, and *cyber security* refer to threats and conflicts in the cyber space, either with military or criminal background, by means of IT. Very often, however, it is almost impossible to recognize the intentions behind the attack (either military or criminal), since attackers are mostly using botnets (i.e., networks of computers that are infected by malware and under the control of cyber criminals). In fact, there is always a battle between the latest attack methods, on the one hand, and the protective mechanisms, on the other side.


The challenges are increased by the fact that a paradigm shift can be recognized with respect to the targets being attacked. So far, mostly state institutions have been the target of attacks. Nowadays, an increasing number of attacks against other targets, such as specific industrial or other enterprises as well as organizations, are recognized. Stuxnet is a recent example of a malware – a computer worm – that targets only SCADA systems.

The increasing usage of encrypted data, the number of targeted attacks, the mobility, the heterogeneity, as well as the need to detect insiders, are demanding challenges for cyber defense. Current approaches for detecting attacks are not sufficient. Signature-based approaches, where the collected data is compared to known patterns (signatures), the most widespread and used approach in intrusion detection/prevention systems (IDS/IPS), as well as virus scanners, are not suitable to detect targeted attacks. Anomaly-based approaches that observe the behavior of the system – instead of searching for patterns – to identify anomalies are much more promising, however, not yet really applicable.

Since IT security needs to be addressed in a holistic way, it is necessary to address aspects raising from the protection of communication infrastructures (systems, services, and data), critical infrastructures, such as energy networks, to cloud services and cloud resources. The research activities of Cyber Defense@UniBwM are focusing on these topics.

3.15 The Threat of Love Letters: Detection of Document-based Attacks

Pavel Laskov (Universität Tübingen, DE)


License  Creative Commons BY-NC-ND 3.0 Unported license
© Pavel Laskov

Most of the information that users access via their computers is stored in some non-trivial format, e.g., HTML, PDF, Excel, JPG, etc. Users access this information via appropriate rendering software which, if vulnerable, can be exploited by sending a specially crafted “document” in the respective format. Due to the high complexity of formats as well as of the rendering software, a steady flow of vulnerabilities is discovered which can be potentially exploited before the vulnerabilities are patched. Traditional signature-based methods are hardly adequate for protection against document-based attacks, since they mostly detect only old attacks with well-known signatures.

In this talk, the challenges of detecting novel attacks that are embedded in specific document formats are elucidated. The talk contains a presentation of previous relevant work on detection of attacks using embedded JavaScript and a discussion on its features and limitations. Finally, a new approach pursued in our current work for building a general framework for detection of document-based attacks is outlined.

3.16 Learning-based Defenses against Malicious JavaScript Code

Konrad Rieck (Universität Göttingen, DE)


License  Creative Commons BY-NC-ND 3.0 Unported license
© Konrad Rieck

Joint work of Rieck, Konrad; Krueger, Tammo; Dewald, Andreas
Main reference K. Rieck, T. Krueger, A. Dewald, “Cujo: Efficient Detection and Prevention of Drive-by-Download Attacks,” in Proc. of 26th Annual Computer Security Applications Conference (ACSAC’10), pp. 31–39, ACM, 2010.
URL <http://dx.doi.org/10.1145/1920261.1920267>

JavaScript is increasingly used for exploiting vulnerabilities in web browsers and infecting users with malicious software. Conventional detection systems that rely on rules and signatures fail to protect from these attacks, as they are unable to cope with the evolving diversity and obfuscation of malicious JavaScript code. This talk explores how machine learning can be applied for analyzing and identifying JavaScript attacks more effectively. Different approaches from recent research are presented along with empirical results and perspectives for future work.

3.17 Semantic Exploration of DNS

Radu State (University of Luxembourg, LU)


License  Creative Commons BY-NC-ND 3.0 Unported license
© Radu State

The DNS structure discloses useful information about the organization and the operation of an enterprise network which can be used for designing attacks as well as monitoring domains supporting malicious activities. This talk introduces a new method for exploring

the DNS domains. In contrast to our previous work of a tool to generate existing DNS names accurately for probing a domain automatically, the presented approach is extended by leveraging the semantic analysis of domain names. In particular, the semantic distribution similarities and relatedness of sub-domains are considered as well as sequential patterns. The evaluation shows that the discovery is highly improved, while the overhead remains low compared to non-semantic DNS probing tools including ours and others.

3.18 Intrusion Detection for the Web 2.0

René Rietz (BTU Cottbus, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© René Rietz

The talk is about the limitations of intrusion detection systems (IDS) in the context of the Web 2.0. Most classical IDS have been designed with simple buffer overflows in mind, but they do not work if they face structured web content with plenty of opportunities for obfuscation. We think that the protection of the clients and servers of Web 2.0 applications requires some kind of firewall approach which denies or allows specific web applications. For this we have to analyze the web languages (HTML, XML, JavaScript, ...) and to describe the structure of their contents. If we can identify these structures later, it is possible to pass or deny the underlying packets in a web firewall and to detect any changes (e.g., malicious code).

3.19 From the Unexpected Side: SkyNET


Sven Dietrich (Stevens Institute of Technology, US)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Sven Dietrich

The talk considers attacks that bypass the traditional sensor/IDS locations. By using a commercially available toy drone, it is possible to compromise wireless access points and to install a botnet. The network of bots is separated from the botmaster. Linkage between botnet and botmaster is realized by one or more drones. After presenting the differences of such a scenario compared to the traditional botnet behaviour the attack framework is explained. Finally, the challenge for network security in such a scenario is outlined. Since the proximity to the target is easy to realize, a new stance for network defense has to evolve.

3.20 State of the Art and Limitations to Application Level Firewalling and Virus Scanning

Alexander von Gernler (GeNUA – Kirchheim bei München, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Alexander von Gernler

This talk gives a presentation of the current practice of application level firewalling and virus scanning from the perspective of a firewall manufacturer.


Application level firewalling means the interpretation and possibly normalization or modification of traffic passing through the firewall. It is performed at OSI layers higher than 4. While being more expensive, it filters out certain connection-based attacks implicitly and allows for mitigating other attacks easily, also by scanning processed content for viruses on the fly.

Virus scanning in this case cannot be given solely to the usual suspects, the anti-virus industry, but is helped by the firewall by doing preprocessing. As with honeypots and the anti-virus software itself, the way that the firewall interprets content is highly relevant for the security of the whole system.

The talk is concluded by presenting a relatively new problem that emerged with the rise of Web 2.0 applications: Suddenly, web content to be processed by the firewall is no longer static, but carries executable content (here: JavaScript). As many web-based attacks rely on malicious JavaScript code, Web 2.0 applications represent a whole new attack vector. The talk ends with a discussion of some thoughts about this new phenomenon and open questions regarding the topic.

3.21 Bro 2.0 and Beyond

Robin Sommer (ICSI – Berkeley, US)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Robin Sommer


In this talk the most recent version of Bro, an open-source network security monitor, is presented that has been developed in our group at ICSI for more than a decade now.

Today, Bro is used operationally by many universities, labs, and science communities to protect their infrastructure. The talk starts with a short introduction to Bro focusing on the differences between Bro and other systems in the same space. Then, the main changes going into Bro 2.0 are summarized, and the roadmap for the near-term future is presented.

In the second part of the talk, two areas that Bro-related research at ICSI is currently focusing on are discussed: (1) integrating real-time intelligence into the system and (2) increasing performance to address emerging 100 Gbit/s deployments.

3.22 A Dynamically Adapting Distributed Multi-Agent IDS

Michael Vogel (BTU Cottbus, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Michael Vogel

Joint work of Vogel, Michael; Schmerl, Sebastian; Schuster, Franka
Main reference M. Vogel, S. Schmerl, H. König, "Efficient Distributed Signature Analysis," in Proc. of the 5th Int'l. Conf. on Autonomous Infrastructure, Management, and Security (AIMS'11), pp. 13–25, LNCS, vol. 6734, Springer, 2011.
URL http://dx.doi.org/10.1007/978-3-642-21484-4_2

This talk is motivated by the problem current IDS have to face because of the growing gap between the evolution of network bandwidth and the single-thread performance of today's CPU architectures. Especially in the case of high analysis load caused by network traffic characterized by short network flows in average and many contained attack traces, the Snort

IDS throughput cannot keep pace with the link bandwidth, so that monitoring data has to be dropped.

In this talk a dynamically adapting and distributed intrusion detection infrastructure is proposed which can utilize different existing IDS as a black box. Resource shortages in high bandwidth situations can be handled by analysis distribution. Performance improvements that could be gained by function or data parallel approaches are discussed for an existing multi-step signature-based IDS as well as the Snort IDS (single step signatures). Finally, the architecture of an IDS agent is presented that applies the examined distribution approaches to dynamically adapt to changing analysis demands and available resources.

3.23 Security Challenges of IPv6 Networks

Bettina Schnor (Universität Potsdam, DE)


License  Creative Commons BY-NC-ND 3.0 Unported license
© Bettina Schnor
URL <http://www.idsv6.de>

The transition from IPv4 to the official successor protocol IPv6 is on the way. New features like for example MTU path discovery have to be supported by IPv6 firewalls with new filter rules. IPv6 comes along with new concepts like the stateless address autoconfiguration (SLAAC) which results in new ICMPv6 message types and new security risks.

There is still a deficit of tools for the analysis of the threat level in IPv6 networks. The same applies to the testing of IPv6 firewalls and intrusion detection systems. The talk presents some of the IPv6 security risks and gives an overview over the IDSv6 project: the Snort IPv6 extension, which detects attacks on the IPv6 address autoconfiguration, and the honeypot honeydv6.

3.24 Anomaly Detection in Challenged Networks

Simin Nadjm-Tehrani (Linköping University, SE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Simin Nadjm-Tehrani

This talk addresses information dissemination in disaster area networks with common handheld devices using no existing infrastructure. The open and distributed nature of these networks makes them challenging from the security point of view. Malicious actors may try to disrupt the communication to create more chaos for their own benefit.


The talk presents a general survivability framework for monitoring and reacting to disruptive attacks. The idea is to have a fully distributed framework to detect anomalies, diagnose them, and perform mitigation individually in each node while adapting to the changing environment.

The approach has been evaluated in the context of a simulated disaster area network running a manycast dissemination protocol that uses the Wifi interface in ad hoc mode. The results demonstrate that the approach diminishes the impact of attacks considerably. In addition, in order to evaluate the impact on the resources and specifically the energy footprint of the survivability framework itself, the framework has been implemented in a modular way in an Android smart phone.

4 Working Groups

4.1 Breakout Session: Future of Early Warning Systems

Bettina Schnor (Universität Potsdam, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Bettina Schnor

The participants of the breakout session came from research institutes and companies developing or operating EWS components. First, the group discussed an appropriate definition of EWS and the distinction from classical IDS. The group agreed that the following characteristics are essential: An EWS monitors *large-scale* networks and also does prediction.

There was a longer discussion whether an EWS is comparable with a weather forecast. The group came to the conclusion that this comparison is not feasible, since an EWS cannot make predictions like “There will be an attack starting in four hours.” Instead, some machines will get infected and after identifying the attack, there will be a prediction about the current network situation and the expected propagation. Also the group agreed that a recommendation of actions is not in the scope of an EWS. Further, an EWS will *not* detect a targeted attack.

It was stated that the target group is limited and that the government is very much interested in EWS, since it is responsible to maintain the cyber infrastructure. The demand is not only an “early warning”, but also to provide “situational awareness”, i.e., the EWS should help to answer questions like: What is happening and why?


A DFN-CERT member reported about the experience in operating the early warning system CarmentiS (<http://www.carmentis.org>). According to that, the interpretation of the results requires a security expert and cannot be automated. The experience with CarmentiS also shows that seeing only a small portion of the network tends to give a representative impression of the whole network. Even for EWS approaches like CarmentiS, it is hard to enroll data suppliers. The reason for this is not technical, but political/psychological. There are some patterns of attacks known/understood. Hence, it should be possible for an EWS to give hints to understand even *new* attacks.

The group collected the expectations that an EWS should warn as early as possible, predict infection propagation (“Ausbreitungsphänomene”), warn about DDoS and massive client-side attacks, and give automatically generated hints on further investigation of *new* attacks/malware like IP ranges and ports. Furthermore, it should perform a multi-layer analysis approach by providing data analysis at different abstraction levels.

Finally, open questions were collected: Do we need new methods for data correlation? What are the challenges for EWS in IPv6/mobile/wireless networks? What data is of special interest for an EWS? What data should be provided for “situational awareness”?

4.2 Breakout Session: Cloud Security

Holger Kinkel (TU München, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Holger Kinkel





The breakout session on “Cloud Security” attracted seven people. It turned out that cloud security is a very broad field of discussion. Problem fields need to be classified according

to the type of the cloud service (infrastructure/platform/software as a service), to the level of privacy (private/hybrid/public cloud), and to technical and non-technical (e.g., legal or organizational) aspects. Thus, the expectations of the participants on the subjects to discuss were quite divergent.

One of the main discussion points was the question if there are any new research challenges regarding cloud security or whether existing solutions only may be adapted to the cloud. The discussion was quite controversial and without a clear result. The impression was that some research questions are common, others specific. For instance, challenges regarding cloud forensics (i.e., which evidence needs to be secured for criminal prosecution, the server or the virtual machine only) or how to control where data is allowed to be stored or processed in the cloud (i.e., tags on data items define where the data can go to, etc.) seem to be specific.

4.3 Breakout Session: Teaching IT Security


Hervé Debar (Télécom SudParis, Evry, FR)

License     Creative Commons BY-NC-ND 3.0 Unported license
© Hervé Debar

The breakout session on “Teaching IT Security” gathered six people, all active in running IT security-related curricula. The group had a round-table discussion on the practices of each of the represented institutions, teaching network and systems security at bachelor and master level. In a nutshell, there is a convergence in the programs taught and the course material used at the institutions represented in the session.

4.4 Breakout Session: Information Security for Novel Devices

Elmar Gerhards-Padilla (Universität Bonn, DE)


License     Creative Commons BY-NC-ND 3.0 Unported license
© Elmar Gerhards-Padilla

The breakout session on “Information Security for Novel Devices” came to the insight that you need to take into account the characteristics of new devices when thinking about information security for these devices. The characteristics relevant in this context are: platform and software diversity, resource restrictions, criticality, and level of interaction. These characteristics have a significant impact on the applicability of conventional network attack detection and response mechanisms.

Anti-virus components are ruled out largely by resource and platform limitations, while EWS are limited mainly to broad-based attacks. Thus, IDS/IPS involving future devices seem to be the most promising field of research for information security on novel devices.

4.5 Breakout Session: Fighting against Botnets

Michael Meier (TU Dortmund, DE)


License  Creative Commons BY-NC-ND 3.0 Unported license
© Michael Meier

The breakout session on “Fighting against Botnets” attracted about ten people having very different expectations of the session. During the round-table discussion legal and social issues have been identified as important, but a discussion of these issues was postponed to another breakout session.

Besides some technical questions, for which the recent botnet study by ENISA was referred to, the more controversial question was whether botnets are still a problem. The main results of the overall discussion can be summarized as follows: Botnets are still a problem which will last forever and they will adapt to new (currently mobile) devices. For successfully fighting against botnets, a number of legal, social, and ethical questions have to be answered.

4.6 Breakout Session: Smart Energy Grids

Michael Vogel (BTU Cottbus, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Michael Vogel

Joint work of Bueschkes, Roland; Herrmann, Peter; Nadjm-Tehrani, Simin; State, Radu; Vogel, Michael; Wolthusen, Stephen

The breakout session on “Smart Energy Grids” was formed by six participants. The discussion started by identifying the components and layers of today’s energy grids and future smart energy grids. Then, security measures and possible attack vectors of smart grids have been discussed.

Finally, the participants identified preliminary security requirements and necessary research areas and gave two summarizing hypotheses on security measures for future smart energy grids: (1) preventive security in smart grids (e.g., the use of meters) is expensive and (2) reactive measures (e.g., anomaly detection) are mandatory.

4.7 Breakout Session: Critical Infrastructure Protection

Franka Schuster (BTU Cottbus, DE)


License  Creative Commons BY-NC-ND 3.0 Unported license
© Franka Schuster

The breakout session on “Critical Infrastructure Protection” attracted more than ten people with very different previous knowledge about the topic. Thus, right at the start the question was raised, why critical infrastructures are so difficult to protect. It was stated that the introduction of IT into plant administration, control, and maintenance connects former isolated systems to the Internet world. Hence, the group agreed that security measures have to be developed which can be non-invasively applied to existing complex and highly-tailored industrial implementations with respect to real-time constraints.

The further discussion focussed on the determination of scientific research challenges on that field. The need for authentication protocols, special cryptographic protocols, and anomaly detection considering the infrastructural context was identified. Finally, specific threats and risks for critical infrastructures were made part of the session, and the limits of such threat and risk analysis were estimated.

4.8 Breakout Session: Cyber-Physical System Security

Hartmut König (BTU Cottbus, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Hartmut König

Recently, cyber-physical systems have been identified as a key research area in the years to come. These are systems which possess an intense link between the computational and physical elements. Input and output are usually realized via the physical elements. The use of the term cyber-physical system, however, is still vague. There exist many different definitions which overlap with other areas, e.g., with that of critical infrastructures. It has something of a buzzword with still varying interpretations behind it.

The development of cyber-physical systems comprises a broad range of scientific challenges [1] covering many areas which have been investigated in computer science already for years. Security is mentioned as one of the key issues [2]. The group agreed that the understanding of security challenges, however, is not matured, yet. Several ad hoc discussion papers indicated some research directions.

The further discussion pointed out that attacks on cyber-physical systems may be more complex than on IT systems; their impact may be larger. On the other hand, the security challenges of cyber-physical systems have not precisely defined up to now as well as their relation to privacy and anonymity. For that reason, the group proposed to apply for a Dagstuhl perspective workshop on the security challenges of cyber-physical systems.

References

- 1 M. Broy (Hrsg.): Cyber-Physical Systems: Innovation durch Software-intensive eingebettete Systeme. Springer, 2011
- 2 P. Pal, R. Schantz, K. Rohloff, J. Loyall: Cyber-physical Systems Security – Challenges and Research Ideas. Workshop on Future Directions in Cyber-physical Systems Security 2009

Participants

- Nils Aschenbruck
Universität Bonn, DE
- Lothar Braun
TU München, DE
- Roland Büschkes
RWE IT GmbH – Essen, DE
- Georg Carle
TU München, DE
- Hervé Debar
Télécom SudParis – Evry, FR
- Sven Dietrich
Stevens Inst. of Technology, US
- Till Dörge
PRESENSE Technologies GmbH
– Hamburg, DE
- Gabi Dreö Rodosek
Universität der Bundeswehr –
München, DE
- Falko Dressler
Universität Innsbruck, AT
- Ulrich Flegel
University of Applied Sciences –
Stuttgart, DE
- Felix C. Freiling
Univ. Erlangen-Nürnberg, DE
- Elmar Gerhards-Padilla
Universität Bonn, DE
- Peter Herrmann
NTNU – Trondheim, NO
- Marko Jahnke
Fraunhofer FKIE –
Wachtberg, DE
- Holger Kinkel
TU München, DE
- Hartmut König
BTU Cottbus, DE
- Jan Kohlrausch
DFN-CERT Services GmbH, DE
- Pavel Laskov
Universität Tübingen, DE
- Corrado Leita
Symantec Research Labs –
Sophia Antipolis, FR
- Michael Meier
TU Dortmund, DE
- Simin Nadjm-Tehrani
Linköping University, SE
- Andreas Paul
BTU Cottbus, DE
- Aiko Pras
University of Twente, NL
- Konrad Rieck
Universität Göttingen, DE
- René Rietz
BTU Cottbus, DE
- Sebastian Schmerl
AGT Germany – Berlin, DE
- Bettina Schnor
Universität Potsdam, DE
- Franka Schuster
BTU Cottbus, DE
- Robin Sommer
ICSI – Berkeley, US
- Radu State
University of Luxembourg, LU
- Jens Tölle
Fraunhofer FKIE –
Wachtberg, DE
- Michael Vogel
BTU Cottbus, DE
- Alexander von Gernler
GeNUA – Kirchheim bei
München, DE
- Stephen Wolthausen
RHUL – London, GB



Software Clone Management Towards Industrial Application

Edited by

Rainer Koschke¹, Ira D. Baxter², Michael Conradt³, and James R. Cordy⁴

1 Universität Bremen, DE, koschke@informatik.uni-bremen.de

2 Semantic Designs – Austin, US, idbaxter@semdesigns.com

3 Google – München, DE, conradt@google.com

4 Queens University – Kingston, CA, cordy@cs.queensu.ca

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 12071 “Software Clone Management Towards Industrial Application”. Software clones are identical or similar pieces of code or design. A lot of research has been devoted to software clones. Unlike previous research, this seminar put a particular emphasis on industrial application of software clone management methods and tools and aimed at gathering concrete usage scenarios of clone management in industry, which will help to identify new industrially relevant aspects in order to shape the future research.

Talks were presented by industrial participants and working groups were formed to discuss issues in clone detection, presentation, and refactoring. In addition we developed a unified conceptual model to capture clone information required to support a common notion of clone data and for interoperability to foster exchange of data among researchers and tools in practice. The main focus of current research is clones in source code – therefore, we also looked into ways of extending our research to other types of software artifacts. Last but not least, we discussed how clone management activities may be integrated into the process of software development.

Seminar 12.–17. February, 2012 – www.dagstuhl.de/12071

1998 ACM Subject Classification D.2.7 Distribution, Maintenance, and Enhancement, D.2.13 Reusable Software, K.5.1 Hardware/Software Protection

Keywords and phrases Software clones, code redundancy, clone detection, redundancy removal, software refactoring, software reengineering, plagiarism detection, copyright infringement, source differencing

Digital Object Identifier 10.4230/DagRep.2.2.21

1 Executive Summary

Rainer Koschke (University of Bremen, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Rainer Koschke

Software clones are identical or similar pieces of code or design. They are often a result of copying and pasting as an act of ad-hoc reuse by programmers. Software clone research is of high relevance for software engineering research and practice today. Several studies have shown that there is a high degree of redundancy in software both in industrial and open-source systems. This redundancy bears the risk of update anomalies and increased maintenance effort.



Except where otherwise noted, content of this report is licensed under a Creative Commons BY-NC-ND 3.0 Unported license
Software Clone Management Towards Industrial Application, *Dagstuhl Reports*, Vol. 2, Issue 2, pp. 21–57
Editors: Rainer Koschke, Ira D. Baxter, Michael Conradt, and James R. Cordy



Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Many techniques exist that try to detect clones. Some of them are already available in open-source (e.g., PMD) as well as commercial tools (e.g., CloneDr). There are also lines of research in clone detection that evaluate these approaches, reason about ways to remove clones, assess the effect of clones on maintainability, track their evolution, and investigate root causes of clones. Today, research in software clones is an established field with more than 100 publications in various conferences and journals.

The purpose of this seminar was to solidify and give shape to this research area and community. Unlike previous similar events, this Dagstuhl seminar put a particular emphasis on industrial application of software clone management methods and tools and aimed at gathering concrete usage scenarios of clone management in industry, which will help to identify new industrially relevant aspects in order to shape the future research. Research in software clones is very close to industrial application. Among other things, we focused on issues of industrial adoption of our methods and tools.

To achieve our goals, we invited many participants from industry. We managed to reach a percentage of about 30 % industrial participation. Talks were given mostly by industrial participants who shared their experiences with us and posed their problem statements. Academic participants were allowed to give a talk if their talk had a clear focus on industrial experiences, needs, problems, and applications of software clone management and related research fields. The focus, however, was on interaction in form of plenary discussions and smaller working groups. The topics for working groups were gathered by clustering issues the participants wanted to discuss at the seminar. The seminar wiki was used intensively to record the results of the working groups. This agile format was very much appreciated by the participants.

The following working groups were formed:

- **Detection/Use cases:** This working group discussed issues in detecting clones. Because there are already many clone detectors, the focus of this working group was to gather use cases for these. The particularities of a use case dictates what kinds of features a suitable clone detector should have.

The group's result was a list of different use cases for clone detection and an enumeration of distinct features a clone detector should have to support the respective use case. An overview of known limitations and issues of actual clone detectors is also provided along with some research questions oriented towards the improvement of clone detection techniques.

- **Presentation:** Because clone detectors typically find many clones in large systems, the user faces a huge amount of data he or she needs to make sense of. Visualization is a means of presenting large and complex data that takes advantage of a human's ability for visual pattern matching. This working group dealt with presentation issues of clone information. Again, use cases were enumerated because suitability of visualization is task dependent.

The group connected the identified use cases with different existing types of software visualization suitable for these.

- **Interoperability:** To foster collaboration among researchers it is helpful to build interoperable tools. Then, for instance, the result of one researcher's clone detector could be fed into the visualization tool of another researcher. Interoperable tools are also needed to serve practitioners' diverse needs.

This working group created a common model to represent clone information that addresses the needs of a wide range of use cases in research and practice.

- Refactoring: Contrary to the abundance of available clone detectors, there are relatively few tools that help in removing clones. The purpose of this working group was to consider the mechanics and utility of forming clone abstractions and achieving clone refactoring. The group identified various means of eliminating clones that are either provided by the languages the clones are written in or by abstraction outside of the language (e.g., code generation). It also delved into managerial aspects of clone refactoring and particularities of clones in software product lines.
- Clone management (process): Clone management is the set of activities to detect, track, assess, handle, and avoid clones. This working group went into the matter of where clone management may play a role in the development and maintenance process. The group discussed how clone analysis fits into the overall software development process (requirements engineering, development, testing, after deployment). They broached the issue of relation of code search and clone detection and how clone detection could be used in recommender systems.
- Provenance and clones in artifacts that are not source code: Most research in software clones focuses on source code, but as it has been shown by several researchers, clones can also be found in other software artifacts such as models and requirement specifications. This working group investigated needs to extend our research into these fields and the particularities of these fields with respect to clone detection. In addition to that, this working group dealt with provenance of clones, that is, the question where the clone comes from. Although the issues of provenance and clones in other artifacts appear to be largely independent, this working group worked on them jointly for organizational issues. The group elaborated how clones could be detected and handled in binaries, models, and bug reports.

For the remainder of this report, it is important to know the following current categorization of clones:

- Type-1 clone: Identical fragments only.
- Type-2 clone: Lexically identical fragments except for variations in identifiers, literals, types, whitespace, layout, and comments
- Type-3 clone: Gapped clones, that is, clones where statements have been added, removed, or modified.
- Type-4 clone: Semantic clones, that is, clones with similar semantics but different implementations in code.

2 Table of Contents

Executive Summary

<i>Rainer Koschke</i>	21
---------------------------------	----

Overview of Talks

Reducing ROM Consumption by Unifying Clones in Safety-Critical Software Systems <i>Gunther Vogel</i>	25
Code Clone Detection Experience at Microsoft <i>Yingong Dang</i>	25
Clones @ Bosch <i>Jochen Quante</i>	25
Semantic Designs' experience <i>Ira Baxter</i>	26
Clone Detection @Google <i>Michael Conradt</i>	27
Industrial Clone and Malware Detection <i>Andrew Walenstein</i>	27
Where is the “business” case for software clones? <i>Serge Demeyer</i>	27
A Controlled Experiment on Software Clones <i>Jan Harder</i>	28
Issues in detecting license violations <i>Armijn Hemel</i>	29
Good and Evil clones <i>Angela Lozano</i>	29
Improving Software Architecture – Role for Software Clones <i>Ravindra Naik</i>	29

Working Groups

Working group on clone detection <i>Thierry Lavoie</i>	31
Working group on clone presentation <i>Sandro Schulze, Niko Schwarz</i>	35
Working group on interoperability <i>Cory Kapser, Jan Harder, Ira Baxter, Douglas Martin</i>	38
Working group on refactoring <i>Ira Baxter</i>	43
Working group on clone management (process) <i>Jens Krinke</i>	51
Working group on provenance and clones in artifacts that are not source code <i>Serge Demeyer</i>	53




Participants	57
-------------------------------	----

3 Overview of Talks

The seminar asked for lightning talk (a short and intensive talk, typically 5–15 minutes long) on industrial experiences, needs, problems, and applications of software clone management and related research fields. The goal of such talks was to trigger plenary discussions on open, industrially relevant issues rather than to provide found solutions. These problem statements were used during the seminar as work items for the working groups.

3.1 Reducing ROM Consumption by Unifying Clones in Safety-Critical Software Systems





Gunther Vogel (Robert Bosch GmbH, DE)

License     Creative Commons BY-NC-ND 3.0 Unported license
© Gunther Vogel

This talk summarized experiences with clone management during software development of airbag software at Robert Bosch GmbH.

3.2 Code Clone Detection Experience at Microsoft





Yingong Dang (Microsoft Research Asia, CN)

License     Creative Commons BY-NC-ND 3.0 Unported license
© Yingong Dang

This talk presented a clone detector developed at Microsoft Research Asia and some of the experiences gathered in using it within Microsoft.

3.3 Clones @ Bosch


Jochen Quante (Corporate Research at Robert Bosch GmbH, DE)

License     Creative Commons BY-NC-ND 3.0 Unported license
© Jochen Quante

This talk explained clone detection/management activities at Bosch Corporate Research. It stated reasons for clones in Bosch automotive software and discussed their pros and cons. Beyond source code, the talk delves into clones in models of model-driven development. Finally, challenges from Bosch's perspective were listed.

3.4 Semantic Designs’ experience

Ira Baxter (Semantic Designs, US)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Ira Baxter

The Dagstuhl Seminar focused on industrial application of clone detection and management methods, tools, and consequences. Ira Baxter of Semantic Designs built one of the earlier clone detection tools, CloneDR, based on matching abstract syntax trees, and has offered this tool as a commercial product for a decade. This talk sketched Semantic Designs’ scalable program analysis and transformation infrastructure, DMS, and described how CloneDR leveraged the DMS machinery to implement an industrial strength clone analysis tool. DMS’s ability to handle many languages, and its regular architecture, enables CloneDR to be implemented as a product line parameterized by language front ends; clone detectors for new languages can be constructed in about 15 minutes of effort once a language front end for DMS is completed. Notably, across many different computer languages (C, C++, COBOL, Java, Python, PHP and a variety of others), CloneDR consistently finds 10-20 % of the code is cloned. An “impossible software growth” curve with negative growth over time was exhibited for a customer company applying clone removal manually but regularly based on CloneDR analyses. The talk exhibited the HTML report generated by CloneDR, including summary pages and pages shows specific clones. It was a surprise to the author that CloneDR’s presentation of parameterized clones and the bindings for the parameters was not standard.

Experience with clone detection has shown variety of nonstandard uses: a) cherry picking of very large clones is easy and valuable; b) isolating a clone makes the code block easier to understand than when it exists in its surrounding code context, c) if bindings of a clone parameter are of inconsistent conceptual types, the clone is often buggy; d) clone abstractions form the basis for domain concepts and realizations, e) there is considerable utility in applying clone detection to DSLs who themselves often have weak abstraction abilities, to determine the kinds of abstractions that might be useful for that DSL. Finally, the complement of clone detection (“what code is the same”) leads to a focus on “what code is different”, showing a connection between the machinery needed for clone detection and “smart differencing” over ASTs. Semantic Designs has built a product line “Smart Differencers” following this philosophy, and using much of the same machinery. It was suggested that CloneDR might be useful in constructing product lines from forked code bases.

Technology application has proven difficult. The business case for clone detection and removal is not yet clear and management will generally not commit with such business case. Programmers also resist; a) while it is well known that code contains many clones, revealing them shows often embarrassing cloning on the part of individual programmers, b) the absence of IDE integration in their favorite IDE is a significant stumbling block; IDE integration must become a product-line; c) the resistance to “not a free tool” is astonishing considering the value of programmer time. Better models of ROI need to be developed to overcome guesswork about value.

Future developments include better clone detection but perhaps more importantly actual clone removal. Removal requires selection of a specific abstraction method for each subset of a clone set, chosen from both language-supported capabilities (subroutines, macros, etc.), and extra-language capabilities such as general macro processors, configuration conditionals and even wholesale file replacement. The variety of choices here, and the sheer volume of clones to be potentially removed, is a barrier to application because of the level of user effort required. Actual removal requires the ability for an engine to reliably modify the code

according to the abstraction type; as a program transformation engine, DMS is peculiarly well placed for this task, and there are few other practical alternatives. Perhaps integration into an IDE, with “single click to orbit” removal of clones will change to perceived and actual value.

References

- 1 I.D. Baxter, C. Pidgeon, and M. Mehlich. DMS: Program Transformation for Practical Scalable Software Evolution. in *International Conference on Software Engineering*, pp. 625-634, 2004.
- 2 I.D. Baxter, A. Yahin, L. Moura, M. Sant’Anna, and L. Bier. Clone Detection Using Abstract Syntax Trees. In *International Conference on Software Maintenance*, IEEE Press, 1998
- 3 <http://www.semanticdesigns.com>. Semantic Designs Company Website.

3.5 Clone Detection @Google

Michael Conradt (Google, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Michael Conradt

The talk described the experience Google made with clone detection, briefly outlined a few future ideas and what the resulting requirements for a clone detection system are.

3.6 Industrial Clone and Malware Detection


Andrew Walenstein (University of Louisiana at Lafayette, US)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Andrew Walenstein

This presentation looked at commonalities between malware detection and clone detection.

3.7 Where is the “business” case for software clones?

Serge Demeyer (University of Antwerpen, BE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Serge Demeyer

Joint work of Van Rompaey, Bart; Du Bois, Bart; Demeyer Serge et. al.

Main reference B. Van Rompaey, B. Du Bois, S. Demeyer, J. Pleunis, R. Putman, K. Meijfroidt, J. C. Dueñas, B. García, “SERIOUS: Software Evolution, Refactoring, Improvement of Operational and Usable Systems,” in Proc. of 13th European Conf. on Software Maintenance and Reengineering (CSMR’09), pp. 277–280, 2009.

URL <http://dx.doi.org/10.1109/CSMR.2009.30>

Between 2006 and 2008 our research group was involved in the ITEA project entitled SERIOUS (Software Evolution, Refactoring, Improvement of Operational & Usable Systems) [1]. Code Clones as a symptom of refactoring opportunities were of prime importance during this project as the goal of the project was to deliver a refactoring handbook. As such we attempted to establish a so-called “business case” for code clones; that is, we tried to calculate a potential return on investment of refactorings that would remove clones. During


this lightning talk I shared a few anecdotes on this quest for a business case. *SPOILER ALERT*: Unfortunately, the story ends with an anti-climax. In the end, we abandoned the business case for code clones in favour of project-specific business cases.

References

- 1 Bart Van Rompaey, Bart Du Bois, Serge Demeyer, John Pleunis, Ron Putman, Karel Meijfroidt, Juan C. Duenas, and Boni García. Serious: Software evolution, refactoring, improvement of operational & usable systems. In *13th European Conference on Software Maintenance and Reengineering (CSMR 2009)*. IEEE Press, March 2009.

3.8 A Controlled Experiment on Software Clones

Jan Harder (*Universität Bremen, DE*)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Jan Harder

Joint work of Harder, Jan; Tiarks, Rebecca

Main reference J. Harder, R. Tiarks, “A Controlled Experiment on Software Clones,” in Proc. of the Int’l Conf. on Program Comprehension, 2012.


Most software systems contain sections of duplicated source code—clones—that are believed to make maintenance more difficult. Recent studies tested this assumption by retrospective analyses of software archives. While giving important insights, the analysis of historical data relies only on snapshots and misses the human interaction in between. We conducted a controlled experiment to investigate how clones affect the programmer’s performance in common bug- fixing tasks. The experiment is based on two small open-source games FrozenBubble and Pacman. For each system, we defined one maintenance task that requires fixing a bug. For each of these tasks, we prepared two variations that differ only in the independent variable, which is whether the bug is cloned or not. The participants were drawn from two different populations. In total 21 students of the University of Bremen and 12 participants of the Dagstuhl seminar 12071 participated in the experiment. The dependent variables, we observed, were the time needed to fix the bug and the correctness of the solution. The results do not reach statistical significance. Nevertheless, we observed many incomplete bug-fixes—in all cases only the more apparent bug symptom was corrected. When the bug was cloned up to 54.5% of the students failed to fix both locations. But also many of the experts—up to 33.3%—overlooked cloned bugs even though they participated in the context of a clone seminar and should have expected clones. We also observed some differences in the time needed to solve the tasks. In most cases the tasks variants without a clone were solved quicker. In one case, however, the experts were faster fixing the cloned variant. This peculiarity could be caused by the small sample size. A full report on the experiment has been published to ICPC.

References

- 1 J. Harder, R. Tiarks. *A Controlled Experiment on Software Clones*. Proceedings of the 20th International Conference on Program Comprehension, 2012.

3.9 Issues in detecting license violations


Armijn Hemel (GPL Violations Project, NL)

License  Creative Commons BY-NC-ND 3.0 Unported license
 © Armijn Hemel
Joint work of Hemel, Armijn; Vermaas, Rob; Dolstra, Eelco; Kalleberg, Karl Trygve;
Main reference A. Hemel, K. T. Kalleberg, R. Vermaas, E. Dolstra, “Finding software license violations through binary code clone detection,” in Proc. of the 8th Working Conf. on Mining Software Repositories (MSR’11), pp. 63–72, ACM, 2011.
URL <http://dx.doi.org/10.1145/1985441.1985453>

Violations of Open Source licenses such as the GNU General Public License occur very frequently. In this talk the background of violations in the consumer electronics industry was explained, as well as what methods for detection of the presence of Open Source software in unknown opaque binaries, like clone detection, have been successfully applied.

3.10 Good and Evil clones


Angela Lozano (UC Louvain-la-Neuve, BE)

License  Creative Commons BY-NC-ND 3.0 Unported license
 © Angela Lozano

One of the difficulties when considering clone management as part of the quality assurance process is the lack of support for informed decisions on which clones to refactor. Clones are supposed to affect an application on three aspects: they may increase or reduce the changes required by the application, they may help to introduce or avoid bugs, and they may facilitate or hamper the application’s understandability. There are arguments claiming both positive and negative effects on these aspects; but so far, the evidence gathered is not convincing enough to reach an agreement. This presentation aims at increasing the awareness on the importance of discriminating clones, showing some of the limitations of current research, and stating some challenges on separating good from evil clones. Although current findings indicate that only a minority of clones are harmful on the changes that an application requires, they are incapable of distinguishing a-priori which clones would have negative consequences. Ultimately, to allow practitioners to prioritize clone refactorings, clone research should focus on their long-term consequences instead of quantifying their immediate effect.

3.11 Improving Software Architecture – Role for Software Clones

Ravindra Naik (Tata Consultancy Services – Pune, IN)

License  Creative Commons BY-NC-ND 3.0 Unported license
 © Ravindra Naik

The talk presents the problems observed in existing industrial software, primarily business applications, in the context of the role for software clones. For specific problems in migrating towards software product lines, we describe potential solution approaches that can exploit the software clone detection. We describe the problems that were observed with Printer Controller software (engineering application) and Core Banking product (business application). In general our observations are that the enterprise systems are increasingly not able to meet future needs and keep encountering similar function applications in different silos. Some of

the software products, on the other hand, face difficulties in providing new capabilities to all existing customers, and usually customizations (specific to customers) take much longer and are expensive. We note that though exorbitantly expensive, enterprise systems have the option of redesigning and developing from scratch, but the software products do not pragmatically have such an option, lest they are willing to support old customers with versions of old implementation. For software products, migrating to product-line architecture is a potential option [4]. Thus, we observe that software architecture improvement is a common theme across variety of software systems. In the context of software products (especially related to business), we observed that copies are made of the software sources and are customized for every customer. This makes it very difficult for the product team to provide new features to all existing customers, as they have to replicate the new features for every custom implementation. Therefore, among other architecture improvements, migrating to product-line architecture is of prime importance in such cases. Given the situation of one version of the product for each customer and each version having its copy of the source code, the idea is to exploit the capability of Software Clone detection to detect commonality and the variability in the differing assets. The Software Clones in question are a potential variation of the semantic clones or Type-4 clones [2]. Identifying the clones will enable identifying common code, meaning the code that is identical or common in various implementations. Among the variants (which have differing code), identifying the differences, viz. the differing variables, fields, conditional checks, statements, and blocks of code will enable identifying parameters for the variants. Further, the differences need to be detected in functional features or in transactions / processes; there could be constraints under which the differences may (or may not) hold. The critical part of detecting clones is the ability to do so in the presence of multiple functions implemented in a single program or subroutine; also in the presence of already existing but overloaded and inconsistently used parameters [3]. The automation of detection and refactoring, and giving guarantees of the re-factorings are of prime importance for the success of such an approach in the industry.

Acknowledgements

My thanks to various business groups within TCS and my lab head Mr. Arun Bahulkar for the intense discussions and feedback on the software system's problems.

References

- 1 T. Mens and T. Tourwé. 2004. A Survey of Software Refactoring. IEEE TSE 30, 2, 126-139.
- 2 S. Bellon, R. Koschke, G. Antoniol, J. Krinke, and E. Merlo. 2007. Comparison and Evaluation of Clone Detection Tools. IEEE TSE 33, 9, 577-591.
- 3 Hitesh Sajnani, Ravindra Naik, and Cristina Lopes. Application Architecture Discovery – Towards Domain Driven Easily Extensible Code Structure, WCRE Oct. 11, 401-405.
- 4 Angela Lozano. An Overview of Techniques for Detecting Software Variability Concepts in Source Code, ER2011 Workshop – Advances in Conceptual Modelling: Recent Developments and New Directions, Oct. 11, 141-150.

4 Working Groups

During a brainstorming discussion involving all participants, various issues were gathered that should be discussed in separate and parallel smaller working groups. These issues were grouped into cohesive clusters. A working group was formed for each cluster. The identified clusters were as follows (see Section 1 for a short description of their goals and results):

- Detection/Use cases
- Presentation
- Interoperability
- Refactoring
- Clone management (process)
- Provenance and clones in artifacts that are not source code

The following sections summarize the results of these working groups. In two cases – namely, the working groups on *Clone management (process)* and *Provenance and clones in artifacts that are not source code* – we will just report the notes that were added to the seminar’s wiki in the course of the seminar. All other reports are based on the wiki’s entries, too, but were written down and further elaborated after the seminar.

4.1 Working group on clone detection

Thierry Lavoie (*Ecole Polytechnique Montreal, CA*)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Thierry Lavoie

4.1.1 Abstract

Although many efficient clone detectors are readily available, it is still unclear how to use them to solve practical industrial problems. In order to address and focus future research on this issue, many use cases for clone detection were identified and characterised with their defining clone detection features. An overview of known limitations and issues of actual clone detectors is also provided along with some research questions oriented towards the improvement of clone detection techniques.

4.1.2 Introduction

Many clone detection tools are readily available today, but few provide insights on how to interpret and use the detected clones. Even if the state-of-the-art tools have solved the problem of detecting Type-1 and Type-2 clones, many issues need to be addressed both regarding higher types detection and result applicability. In order to propose new focuses for clone detection research, the group identified known issues with current detectors as well as many relevant research questions. As a result, the group suggests to do new clone detection research with a focus on use-case oriented results instead of a broad-scope clone detection.

This report is divided in two sections: the first presents known issues with clone detectors with relevant research questions, and the second presents many use cases and their cloning related features.

4.1.3 Known issues and limitations

Many aspects of type 3 and 4 clone detection are still eluding clone researchers. Those types are required for many use cases. Therefore, it is worth looking at some current problems.

Regarding Type-3 clones, the following questions are still open:

- How can we effectively find Type-3 clones?
- How can we scale Type-3 clone detection effectively?
- Is grouping of Type-3 clones into disjoint sets really appropriate?

With respect to the first question, a use-case oriented approach might suggest a way to better quantify and qualify actual Type-3 clones as it points towards a better clone definition (one that is useful for the use case) instead of the now vaguely defined “gapped” clones. Scalability might as well be solved on a case by case basis. Grouping of clones into disjoint clone classes is a natural choice for type 1 and 2 clones. However, disjoint classes suggest an equivalence relation, which may be ill-formed for Type-3 clones. Specifically, transitivity does not seem to trivially, or at all, hold and symmetry is questionable. Therefore, clone classes should be rethought for Type-3 clones. Regarding Type-4 clones, their current definition as semantic clones is above all too vague. Without a clear conception of what should be a Type-4, or semantic, clone, it is hard to state how it should be detected. Nevertheless, there is a common agreement that only few tools can barely deal with semantic clones and semantic clones are relevant because they do occur in practice.

4.1.4 Limitations of clone detectors

Clone detection tools accuracy still needs improvement. Since the group suggests to head towards use-case-based clone detection, it is natural to ask how can human feedback be used to increase the accuracy of results. Distinguishing relevant and irrelevant clones might become an easier problem if tools are configured for one specific task and results are manually inspected. However, it is still unclear how human feedback might be used meaningfully.

With the evolution of malware and the increase of license infringement problems, obfuscated code becomes an issue for which clone detection tools were not conceived to deal with. Binary clone detection is also relevant for those specific problems and for which tools are not well suited. Investigation of these problems might give potent solution to practical clone detection applications.

4.1.5 Category-oriented use cases

In order to define the challenges modern clone detection tools must overcome to solve practical problems, the group identified several clone detection use cases. For each of them, required features of clone detection tools were identified. In Table 1, the relevant clone types for each use cases are identified. Clone types right to other clone types subsume them. For example, *Type-3 large gap* subsumes *Type-3 small gap*, Type-2 and Type-1 and is itself subsumed by Type-4.

In Table 2, other relevant features are identified. A cross in a cell indicates the feature is required. Each feature is defined as follow:

- Precision: A low rate of false positives is required
- Recall: A low rate of false negatives is required
- Online: A fast, realtime tool is required
- Granularity: The desired size of the clones. *Fine* means small fragments are desired whereas *coarse* indicates the need to identify only large fragments. *Fine&Coarse* indicates clone size is not relevant.
- Incremental: The tool needs to handle multiple fragment additions and deletions
- Blacklisting: The tool needs to handle a corpus of code that must not be considered clone
- Binary: The tool needs to find clones in source code as well as in executable binaries
- Counter-obfuscate: The tools need to deal with obfuscated sources or binaries

■ **Table 1** Highest relevant clone types for identified use cases

Use Case	Type-1	Type-2	Type-3 small gap	Type-3 large gap	Type-4
Abstraction identification			X		
Version analysis tasks				X	
Code reduction		X			
License infringement		X			X
Plagiarism			X		
Code Leakage				X	
Provenance			X		
Productivity measurement		X			
Quality assessment			X		
Regulations complianc		X			
Malware					X
Program comprehension					X
Awareness				X	

■ **Table 2** Required features of clone detection tools for identified use cases

Use Case	Precision	Recall	Online	Granularity	Incremental	Blacklisting	Binary	Counter-obfuscate
Abstraction identification	X			Fine&Coarse				
Version analysis tasks				Coarse	X			
Code reduction	X			Fine&Coarse				
License infringement	X			Coarse		X	X	X
Plagiarism		X		Coarse		X		X
Code Leakage		X		Coarse				
Provenance	X			Coarse	X			
Productivity measurement	X			Fine&Coarse	X			
Quality assessment	X			Coarse	X	X		
Regulations compliance		X		Fine&Coarse				
Malware	X			Coarse			X	X
Program comprehension	X		X	Coarse				
Awareness	X		X	Fine				

4.1.6 Business cases and Cost/Benefits analysis

Using the identified use cases for business purposes is not straightforward. In many cases, a cost/benefits analysis must be first performed to decide whether or not clone analysis is worth investigation. The followings are research questions for which an answer would provide a better intuition on how to use clone detectors in industrial applications:

- What is the business case for clone search and reduction?
- How to measure whether code-clone removal takes less effort than clone management?
- How much does the cost to remove a clone increase with age?
- How can we measure the benefits of clone detection?
- Can we empirically characterize the costs / benefits of different clone refactorings?
- How to get statistics about costs / risks associated with existing clones or avoided clones?
- How to determine the relative importance of clones in a project?

For some use cases, some ways of determining the industrial benefits were identified:

- Abstraction identification: speed-up development by refactoring and having better knowledge of the system
- Version analysis task: speed-up in version merging using clone detection instead of other techniques
- License infringement and provenance: avoid legal problems and reduce costs of legal department
- Productivity measurement: increase in management decision quality
- Quality assessment: reduction in maintenance cost, reduction in audit cost, increased quality of internal assessment, increase quality of third-party quality assessment of suppliers (software escrow)
- Program comprehension: decrease time in comprehension

4.1.7 Conclusion

The group identified many relevant clone-detection use cases along with their required clone-detection features. The group also supports reorientation towards application-oriented clone detection instead of self-purposed-oriented clone detection. In many cases, state-of-the-art clone detection tools do not behave well for these features. These observations point to new research opportunities to enhance clone detection technologies.

4.1.8 Participants

The following people took part in the group discussion and contributed the main ideas of this report:

- Andrew Walenstein, University of Louisiana at Lafayette
- Jochen Quante, Robert Bosch GmbH
- Elmar Jürgens, TU München
- Serge Demeyer, University of Antwerp
- Yingnong Dang, Microsoft Research Beijing
- Stephan Diehl, University Trier
- Jim Cordy, Queens University
- Rainer Koschke, University of Bremen
- Michel Chilowicz, Université Paris-Est

- Thierry Lavoie, École Polytechnique de Montréal
- Werner Teppe, Amadeus Germany GmbH
- Martin Robillard, McGill University
- Rebecca Tiarks, Bremen University
- Michael Conradt, Google
- Minh Zibran, University of Saskatchewan
- Jindae Kim, HK UST

4.2 Working group on clone presentation

SSandro Schulze, Niko Schwarz

License © ⓘ ⊖ Creative Commons BY-NC-ND 3.0 Unported license
 © Sandro Schulze, Niko Schwarz
Main reference S. Schulze, N. Schwarz, “How to Make the Hidden Visible – Code Clone Presentation Revisited,”
 Technical Report FIN-05-2012, University of Magdeburg, Germany, 2012.
URL http://www.cs.uni-magdeburg.de/inf_media/downloads/forschung/technical_reports_und_preprints/2012/04_2012.pdf

4.2.1 Abstract

Nowadays, a slew of clone detection approaches exists, producing a lot of clone data. These data have to be analyzed manually or automatically. It is not trivial to derive conclusions or even actions from the analyzed data. In particular, we argue that it is often unclear how to present the clone information to the user. As a result, we present our idea of task-oriented clone presentation based on use cases. Hence, we propose five use cases that have to be addressed and suggest clone presentation techniques that we consider to be appropriate.

4.2.2 Introduction

Intensive research has been performed on clone detection and evaluation—presentation is often left as an implementation detail to implementors. While there is a plethora of visualizations, current visualization for code clones is limited [1, 2]; they can not serve different issues (e.g., online clone reporting, quality assessment, refactoring). They are rather directed to a certain task for which they are more or less appropriate.

We want to stimulate the topic by discussing what is needed to present and visualize code clones to an end user. This inherently raises the question: What do we want to discover from the code clones, once they have been found by a detection tool? If we can clearly answer this question, we have the ability to find appropriate methods to present this information.

So far, different tasks, related to detected code clones, require different tools to reveal information that is needed for a particular task. In this report, we propose a mapping that shows which visualizations and presentation concepts can serve which purpose. While our suggestions are far from being complete, the objective is to guide tool builders and give an overview over what is there and how it could be exploited. Our vision is a tool or IDE that seamlessly integrates these approaches to provide different views on clones and thus fit the needs of different stakeholders.

4.2.3 Use Cases for Clone Presentation

Once code clones have been detected and analyzed, they must be accessible for further treatment. This step, called clone presentation, is not an obvious task. First of all, there

might be different stakeholders such as software quality managers or software developers that need different views (including different levels of granularity) on the clones. Second, these stakeholders want to perform different actions. In the following, we propose five use cases that encompass the different views and treatments of clones.

4.2.3.1 Quality assessment (QA)

This use case mainly appears on the management level. For instance, the stakeholder wants to have an rough estimation on how the existing clones affect the overall system quality. Furthermore, the detection of hot spots, i.e., parts of a system that contain a larger amount of clones, to define countermeasures or just reason about the clones are part of this use case.

4.2.3.2 Awareness (AW)

This use case describes the fact that it is important for certain stakeholders, especially developers, to be aware of existing clones and how they are related. In particular, during implementation a developer has to know when he changes a cloned fragment. Additionally, the information where the corresponding clones are located is useful to making consistent changes in an efficient way.

4.2.3.3 Bug prediction (BP)

If a bug has been found in a clone of a code snippet, then all other clones might be incorrect as well. Further, if a code snippet is copied from a source to a destination, a certain similarity between source and destination is implied. This could be exploited to predict bug occurrences.

4.2.3.4 Quality improvement (QI)

This use case encompasses persistency and removal of clones. For the first, we envision an enrichment of clone information by the clone producer (i.e., the developer) such as whether a clone is harmful or should not be removed. The latter case encompasses refactoring techniques and all information that is needed to apply them to detected clones.

4.2.3.5 Compliance (CO)

This use case encompasses two issues: First, a stakeholder may be interested in whether code in the systems exists that has been copied from external sources (e.g., third party libraries). Hence, he must ensure that the license is not violated. Second, there could be subsystems that contain code, which is not allowed to be used outside this subsystem such as sensitive code or pre-defined architectural or responsibility boundaries. As a result, it is useful to have a presentation that indicates whether such *internal* compliances are violated.

4.2.4 Putting the Pieces Together

Not all visualizations lend themselves equally to all tasks. In the last section we described the use cases we identified and that have to be addressed by an appropriate clone presentation. However, due to the fact that different approaches are possible for clone representation and visualization, for each use case we focus only on a subset of techniques and methods that we commonly agreed on during intensive discussions. For a more comprehensive overview on possible visualization techniques, we refer to existing surveys on this topic [4, 5]. In Table 3,



■ **Figure 1** Examples for (a) a tree map, (b) a seesoft view, and (c) a compare view from the clone detection and report part of ConQAT [3]

we show a compatibility matrix that relates the use cases to clone presentation methods we propose to address particular use cases.

■ **Table 3** Matrix showing which clone presentation feature can be used for which use case.

	QA	AW	BP	QI	CO
SeeSoft View [6]	X	X	?		X
TreeMap [7]	X	X			X
Source code view		X	X	X	
Compare view			X	X	
Links		X			X
Dashboard	X	?			X
Filtering/querying/zooming	X	X			
User-generated meta-data				X	
Revision history	X		X		

Particularly, we argue that clone visualization such as SeeSoft views or TreeMaps are helpful to provide a big picture of the clones in the system and thus support the use cases QA, AW, and CO. To this end, a SeeSoft view (cf. Figure 1, middle) represents each file as rectangle and each clone as a bar within this rectangle, indicating its size and position. Additionally, code clones that belong to the same clone set have the same color. As a result, the stakeholder receives an overview of clones and how they are scattered throughout the system. Similarly, a TreeMap (cf. Figure 1, left) represents each file as a rectangle with information on size and position, relatively to the whole system. Furthermore, the color indicates whether such a file contains many clones or not, which enables an easy detection of so-called *hot spots*. However, we also propose to make such visualizations more interactive by adding filtering, querying, and zooming capabilities. Particularly for large code bases, this allows to focus on subsets of the overall code base, which are of interest.

In contrast to the previously mentioned visualizations, a developer requires methods for clone presentation that are seamlessly integrated in his development process. We propose that the source code view (as provided by common IDEs) and a compare view (cf. Figure 1, right), providing a face-to-face comparison of two code clones, are appropriate to fulfill these demands and thus to support the use cases BP and QI. For the source code view, we even suggest to integrate more sophisticated approaches such as linking between corresponding clones. As a result, the developer could receive information on corresponding clones in case that he changes a cloned code fragment. Furthermore, he could be provided with means to change the corresponding clones consistently. Beyond that, the compare view can provide even more fine-grained information such as highlighting the differences of two code clones.

Finally, the aforementioned approaches can be complemented by further presentation

techniques. For instance, the revision history can be exploited to provide evolutionary information about the clones while user-generated meta-data (e.g., by *tagging the clones*) can provide useful insights about the developer’s view on certain clones.

4.2.5 Summary

We have summarized the most common use cases of clone detectors and mapped them to visualizations that can display them to the user. While we do not claim completeness, we want to stimulate discussion on our categorization of use cases and the respective clone presentation/visualization approaches.

4.2.6 Participants

Participants of this working group were as follows:


- Hamit Abdul Basit
- Saman Bazrafshan
- Daniel M. German
- Nils Göde
- Martin P. Robillard
- Niko Schwarz
- Sandro Schulze
- Gunther Vogel

References

- 1 R. Tairas, J. Gray, and I. Baxter, “Visualization of Clone Detection Results,” in *Eclipse technology eXchange*. ACM, 2006, pp. 50–54.
- 2 J. Cordy, “Exploring Large-Scale System Similarity Using Incremental Clone Detection and Live Scatterplots,” in *ICPC*, 2011, pp. 151–160.
- 3 E. Juergens, F. Deissenboeck, and B. Hummel, “CloneDetective – A Workbench for Clone Detection Research,” in *ICSE*, 2009, pp. 603–606.
- 4 S. Diehl, *Software Visualization*. Springer, 2007.
- 5 C. K. Roy and J. Cordy, “A Survey on Software Clone Detection Research,” Queen’s University at Kingston, Tech. Rep. 2007-541, 2007.
- 6 S. Eick, J. Steffen, and J. Sumner, E.E., “Seesoft – A Tool for Visualizing Line Oriented Software Statistics,” *IEEE TSE*, vol. 18, no. 11, pp. 957–968, 1992.
- 7 B. Johnson, “TreeViz: Treemap Visualization of Hierarchically Structured Information,” in *CHI*, 1992, pp. 369–370.

4.3 Working group on interoperability

Cory Kapser, Jan Harder, Ira Baxter, Douglas Martin

License  Creative Commons BY-NC-ND 3.0 Unported license
© Cory Kapser, Jan Harder, Ira Baxter, Douglas Martin

4.3.1 Abstract

As the field of code clone research grows, the continuing problem of interoperability between code clone detection and analysis tools grows with it. As a working group, we sought to solve this problem by generating a comprehensive model for code clone detection results that can

be used in a wide range of use cases. As a result, we generated a conceptual model of code clone detection results that can be used to specify exchange languages, web services, output formats, and more. Following the workshop we created an online wiki, where we hope to generate discussion and solidify a shared understanding of the core concepts of the problem domain with the code clone detection and analysis community as a whole.

4.3.2 Introduction

Research on code clones in software – segments of similar code within or between software systems – is continually growing. As the number of code clone detection and analysis tools increases, the number of output formats and parsers for those output formats grows. Yet as a scientific community there is an increasing need to share results, not only for the purposes of replication of experiments, but to enable us to efficiently build on top of each others' results. This leads us to the issue of interoperability of our tools and results.

As a group we realized that before we can solve the problem of interoperability, there needs to be a shared understanding of the core concepts of the problem domain. The working group participants employed object-oriented analysis of the problem domain as a method of identifying important domain concepts. Starting with brain storming use cases for clone detection results, we identified a diverse set of use cases where code clone detection results are used. These became our basis for evaluating the completeness of our concept analysis. Using these use cases as a reference, requirements and core concepts were identified and encoded as classes and associations. The results of this work will continue to evolve, and the most up to date information can be found at <http://www.softwareclones.org/ucm>.

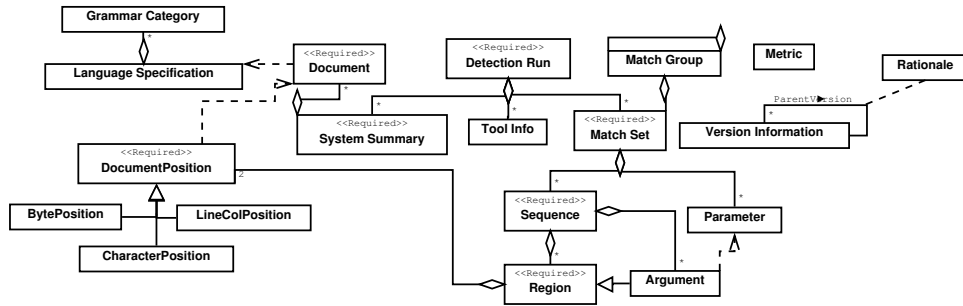
Generic data formats have been proposed [2] but these models may not be complete enough for all available use cases of detection results, nor do they model the core of clone detection results in a truly generic way. Further, these models encode details and constraints specific to their implementations, particularly to suit the models' purpose. For example, RCF specifically models *clone pairs* and *clone classes* separately though it can be argued that the latter is the more general form. Also, the concept of higher level clustering of code clones is not explicitly modelled in RCF. As the model presented here is a description of core concepts and their relationships to one another, potential contributions of this model include:

- a shared decomposition of the problem domain,
- reduced learning overhead for new tool developers and stakeholders as most core concepts have been identified,
- a standardized language for discussing code clone detection results,
- a well defined model to be used to generate a concrete exchange language, and
- a central model for which existing data formats can be documented relative to.

Further, the original use cases can be mapped to the specific concepts in the model, providing a standardized way to communicate minimum requirements for specific usage scenarios.

4.3.3 Use Cases

Three possible domains within which code clone detection would be used can easily be identified: clone detection for computer programming languages, clone detection for non-formal languages (e.g., natural language documents), and clone detection for graph based documents. Working within the first domain during the session, eleven high-level use cases



■ **Figure 2** Unified Clone Model

for clone detection results were identified. These use cases were used to stimulate directed object-oriented analysis going forward as well as verification of the resulting conceptual model afterward. We fully expect this list will be expanded as the larger community is engaged in the discussion. The following use cases were identified:

1. **UC 1: Detect and report.** Detect similarity and simply report it to the user.
2. **UC 2: Detect, report, and track evolution.** Detect similarity and track the evolution of these results across software versions.
3. **UC 3: Detect, report, refactor.** Detect similarity and report them for the purpose of refactoring.
4. **UC 4: Metric analysis.** Generate a metric based analysis of a software system including code clone based metrics (perhaps to study the relationship of code clones, their metrics, and other source code and software development related metrics).
5. **UC 5: Data fusion.** Smarter integration/augmentation of multiple data sources to create more value than the code clone results alone (e.g., improve ROI for code clone analysis by identifying high value/low cost refactoring cases).
6. **UC 6: Scientific replication of a study.** Provide sufficient information about the clones, the detection process, and the source code to replicate the results.
7. **UC 7: Benchmarking.** Benchmark/compare code clone detection tools.
8. **UC 8: Hybrid approaches.** Enable tools to pass data to each other in hybrid clone detection tool chains.
9. **UC 9: Reduce rework.** Provide useful, extra information that could be computed by another tool but presents a significant amount of work. Ensures the results stand completely on their own.
10. **UC 10: Detect for reporting, enable easy navigation and search.** Used to move from clone to clone, snippet to snippet, and enable search within code clones.
11. **UC 11: Plagiarism detection where no source code is available.** May only be able to share minimal results, need to still be able to compare them.

4.3.4 Model

The diagram shown in Figure 2 depicts a model without the concept attributes. In this section, the important features of the model are described and the concept attributes are listed. The conceptual model shown in Figure 2 is encoded as a UML class diagram. Each box represents an important concept identified during the analysis. Those boxes with the stereotype *Required* are deemed to be required for the most basic use case *Detect and Report*. In this case, clone detection results for a single version of the software are simply reported to

the user without any interpretation. As the model reported here reflects the core concepts of the problem domain, it should be noted that this is a model of important concepts, not a data format or OO design.

At its core, the model describes a *detection run* as an instance of a clone detection tool or tools (*tool info*) being run over a document corpus (*system summary*). A detection run also includes a *run start time*. *Tool info* includes the attribute names, version, tool arguments/options, tool chain description, and possibly a boolean to indicate whether or not the code clones detected are returned as classes or pairwise.

The results of the clone detector (code clone pairs or classes¹) are stored as *match sets*. A match set is modelled as a set of *sequences* and *parameters*. Each sequence is an ordered list of *regions*, contiguous segments within a document, that represent the matching fragments detected by the clone detector. The parameters of the match set indicate the points of variability in the mapping (for Type-2 and Type-3 clones this is analogous to gaps in the clone). Each sequence maps an *argument* to each parameter in the match set. In the case of a token based clone detector, a sequence is the whole code fragment that was found to be similar. This sequence is decomposed into the identical fragments (regions) and the differing fragments (parameters/arguments). Match sets, sequences, and regions can have *metrics* and *version information* associated with them as well. This version information could be used to track clones across versions of the document corpus, and versions of sequences across versions of the document corpus. Regions include start and end *Document positions*, text (as found in the system), a checksum, and grammar category (for classification of contained artifacts). Document position representation remains a point of contention within the working group. Three alternatives are suggested in the figure. Byte position, while possibly being the most portable is also the least convenient as that information may be lost in the pre-processing stages. This is similarly true for character position. Line and column position may be the most convenient for many detection tools, but also may require a character interpretation mapping so as to ensure unambiguous interpretation of special characters (such as line feed).

Modelling code clones in this way allows for a sequence of a code clone to span multiple files, and for matching regions to have an arbitrary order (e.g., (1,2,3):(3,1,2)). These scenarios can occur, for example, when clone detectors return results from pre-processed source code where macros have been in-lined [3]. They can also occur when clone detectors that are resistant to line reordering are used, such as PDG based clone detectors [5]. For clone detectors that return clone pairs, a match set would consist of two sequences. For those clone detectors that return clone classes, a match set would contain two or more sequences.

Code clone detectors may apply a clustering of code clones as part of their result set, such as Regional Group of Clones (RGC) [4] or clone classes generated based on the clone pair relationship. In these cases, this can be represented as a *match group*. For higher level clusterings, match groups can also be aggregates of other match groups.

The *system summary* represents a version of the document corpus being analyzed. It consists of *documents*, *metrics*, and *version information*. Documents are the units being analyzed for code clones. Their attributes include the URI, version information, metrics, checksum, original text, preprocessed text, the processed model (such as a serialized AST if one was used) and a *language specification* which provides enough details for the consumer of the detection results to interpret the document position as well as understand how the document was processed by the clone detector. A language specification includes a name

¹ A clone class is a set of two or more code fragments that are considered to match. This is often considered to be an equivalence relation.

(such as C, Java), dialect (such as VS 2008), reference information describing where the language specification can be found, character interpretation to describe how characters map from character or binary offset in the file to line and column positions, and possibly a grammar specification.

Although not shown in the diagram, most concepts can be associated with any number of *metrics*. This is used when additional information, such as similarity, line count, or complexity, are stored. The model is highly extensible in this respect as metrics can be arbitrarily defined using name, value, and type attributes. Also, most nodes can contain *version information*, enabling the tracking of match sets, sequences, and regions independent of versions of the corpus (system summary). This enables, for example, the modelling of genealogies of code clones including linking the origins of specific regions of code. This version information can contain a *rationale* which is used to describe how or why one entity was traced to a prior version.

4.3.5 Conclusion and Future work

The model presented here presents only the beginning of this work. If we wish to create a general model that can be adopted by the community, there needs to be general acceptance by the community. Therefore, a wiki (<http://www.softwareclones.org/ucm/>) has been created to discuss and share the full details of the model. There we will also share the details of reference implementations of database schemas, exchange languages, and web services.


As part of a verification of the basic completeness of the model, a mapping to RCF was performed. The results of this process exposed very few modifications to RCF and no modifications to the model described in this paper. While this is encouraging, we must go further to validate the completeness of the core concepts. In this vain we will perform this mapping to other existing models, including the output of CloneDR [1] but also models not developed by the authors. This will not only ensure we have captured the essence of the problem domain, but also provide examples of how to document existing clone detection result formats relative to this model.

References

- 1 I. D. Baxter, A. Yahin, L. Moura, M. Sant'Anna, and L. Bier. *Clone detection using abstract syntax trees*, in Proceedings of the International Conference on Software Maintenance (ICSM-98), pp. 368, IEEE Computer Society, 1998.
- 2 J. Harder and N. Göde, *Efficiently handling clone data: Rcf and cyclone*, in Proceedings of the 5th International Workshop on Software Clones, pp. 81–82, ACM, 2011.
- 3 I. J. Davis and M. W. Godfrey, *From whence it came: Detecting source code clones by analyzing assembler*, in Proceedings of the 2010 17th Working Conference on Reverse Engineering (WCRE '10), pp. 242–246, IEEE Computer Society, 2010.
- 4 C. J. Kapsner and M. Godfrey. *Improved Tool Support for the Investigation of Duplication in Software*, in Proceedings of the 2005 International Conference on Software Maintenance (ICSM-05), pp. 305–314, IEEE Computer Society, 2005.
- 5 J. Krinke. *Identifying similar code with program dependence graphs*, in Proceedings of the Eighth Working Conference on Reverse Engineering (WCRE '01), pp. 301–309, IEEE Computer Society, 2001.

4.4 Working group on refactoring

Ira Baxter (Semantic Designs, US)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Ira Baxter

4.4.1 Abstract

Much of the research on (software) clones has been focused on methods of detection, understanding, and determining evolutionary properties of clones and their actual impact on software maintenance. However, an implicit assumption behind clone detection is that most or at least some clones should be “refactored” out of existence, unifying the instances into some kind of effective abstraction. Yet there are extremely few tools or methods for actually forming clone abstractions from clones in code or other formal documents, and/or clone refactoring²: replacing cloned artifacts with abstract invocations, and inserting the clone abstraction at some other accessible point in the code.

The purpose of this working group was to consider the mechanics and utility of forming clone abstractions and achieving clone refactoring.

4.4.2 Format

Like the other working groups, we started with index cards containing all-attendees brainstormed one-line topics, that had been filtered into the category that seemed to be “refactoring” (thus the group title). We further classified these into finer sets and tackled each in turn to understand where there might be a synthesis of ideas.

We grouped the topics into several major subtopics, which we discuss below:

- Normal refactoring “within” the formal document
- Using abstractions from outside the language system of the formal document
- Managerial aspects of refactoring: cost, benefits, risks
- Refactoring to product lines

There were several subtopics we did not get to explore and surely deserve the attention of a working group at some future date:

- Clone refactoring applied to non-formal texts (documentation, requirements, parallel refactoring of multiple sets of documents)
- Clone refactoring for graph-structured artifacts, including various types of models
- The relationship of domain analysis/engineering, e.g., clone abstraction to mine reusable components. The observation is that detected clones are often recognized by the programmers that work on a system as to intent, and therefore an abstracted clone has both a concept and a realization, as well as an obvious use in the type of software from which it is extracted.

4.4.3 Clone refactoring: State-of-the-art

At present, most clone detection systems are not associated with any ability to refactor clones for removal (exceptions: CloneDR³ [5], Erlang [12] and Haskell [7], a functional language).

² We suggest using the specific term “clone refactoring” to distinguish this specific activity in the more generic set of activities called “refactoring”.

³ CloneDR was able early in its life to refactor C code with macros, and COBOL code with COPYLIBs. That capability is presently not used.

We considered what technology is available to support clone removal.

The maturity of certain conventional refactorings suggests clone refactoring is practical: “pull up method”, and “form procedure”, both having elements needed for clone refactoring, are generally reliable in participants experience. This suggests that clone refactoring itself should be reliably implementable. There was some contradictory concern that behaviour preserving refactoring is not solved reliably (especially for sequences of refactorings).

One key problem is to obtain robust tools to manipulate the program representation (ASTs, symbol tables, flow analysis), especially for the essentially endless variety of languages for which clone detection seems to be applicable. Most clone analysis tools, e.g., those that match text strings, token sequences, or class files, do not actually have access to such a representation; they have to be integrated with some other tool ecosystem (Eclipse, Clang, ...) to support such refactoring. Clone detectors such a CloneDR [5] built with a general purpose program transformation engine such as DMS [4] should be easier to morph into a clone refactoring tool; such tools have been used to carry out complex code restructurings on languages such as C++ [1].

4.4.4 Clone Refactoring Issues

It is not easy being green. All kinds of issues must be addressed to refactor clones.

- Under the somewhat suspect notion that one wants to remove all clones, can one use an entirely automated approach to remove them? We think this is unlikely: the resulting code is not likely to be understandable, as it is unclear how such a tool would choose a sensible clone abstraction name. Perhaps there are clues in names of variables or in comments or in the nature of the detected clone.
- What are the criteria for suggesting a reasonable refactoring candidate? Can it be tiny (perhaps, if there are hundreds of instances)? Can it have a large number of parameters? Must it be abstractable using a language capability? Should it have some indication of high code churn within the individual clones?
- The right abstraction depends on a lot of information: the parameters (e.g., relationship and count of the different locations), and this seems to be different for each clone. One might desire to refactor (or not!) subsets of a large clone set differently to take advantage of identical parameter bindings. The implications are that removal is likely to be an interactive task.
- Is there only one way to remove a clone? Likely not: several abstractions may be available in the programming language (conditionals, macros, subroutines, ...) for procedural clones, and others available for declaration clones (macros, classes, ...). For any given clone instance, syntax/semantic category, language or client, is there a single preferred way? If so, a clone refactoring tool could have a default method for removal; the user decides if she wants a clone refactored that way. If not, can we provide a catalog of prioritized abstraction possibilities for each detected clone type to help a user choose quickly?
- Is clone removal done only by abstraction capabilities available in the language in which the clone was found, or can one step outside the language and use external metaprogramming techniques to abstract the clone? How does a clone removal tool know about the abstraction methods offered by all the languages it can handle? How does it know about the external metaprogramming facilities?

4.4.5 Structural Clones

Before we discuss abstractions outside the language, we will first examine structural clones [2], as they will play a big role in later discussion on refactoring in this report. The key notion here is that smaller clones may in fact be part of a larger pattern; getters tend to be associated with setters, and so one should expect cloned getters to have corresponding cloned setters. Structural clone detectors use potentially multiple conventional code clone detectors to find clones that form elements of structure clones, and then hunt for repeated patterns of such elements in larger container structures (methods, classes, files, entire directories).

[2] offers one structural clone detector based on item-set frequency analysis. Are there other means to detect structural clones? Can we measure or compare the quality? An open question is “what kind of patterns can be formed from elements to make up a structural clone”? Is the pattern a possible parameter of a structural clone? (e.g., elements $A \dots B$ are found in one container; elements $B \dots A$ are found in another, forming a structural clone with a boolean parameter: “forward order of elements”).

Abstracting structural clones is conceptually very difficult; they may cross many types of language, abstraction and file boundaries. A lot of domain/expert knowledge may be required to abstract a structural clone.

4.4.6 Abstractions outside the language

When refactoring clones, one set of abstraction mechanisms are those offered by the formal language in which the clones were found, e.g., macro and function calls for C, templates and classes for Java, etc. We discussed the idea that a clone refactoring tool might offer refactorings using abstraction mechanism that are not available to that language. A variety of useful generic abstraction/reification mechanisms are available:

- **General macro processors:** One might use (Unix) M4 to supply text-based macro capability to languages that do not have it. (An uglier but similar idea already occurs commonly in large Fortran codes that use the C preprocessor to provide configuration conditionals as well as macros.)
- **Frame generators:** These provide what amounts to tree-structured text macros that generate entire files from explicit configuration parameters driving sophisticated conditionals (Frames [3] or XVCL [10]). GenVoca has been suggested as a generalization of frame technology [6]. We remark that XVCL, being able to produce arbitrary text artifacts, has been used successfully to abstract structural clones.
- **File level selection:** These tools choose between alternatives for files based on configuration conditionals. In essence, these are implicit preprocessor conditionals at the file level. (A product line management tool, Gears [11] offers this as one of its features).
- **Code generators and DSLs:** These generate result code given an input specification in some chosen specification language. A key problem is choosing an appropriate specification language (raising the domain analysis/engineering question), and determining a specification that can be realized to match the clone instances. A special case of this are program transformation systems (e.g., DMS [4]), which can convert abstract operations to target code by applying refinement transformations, and can abstract optimizations as guided sets of transformations. A special case of program transformation systems is intentional programming [13] which might be considered to be a kind of feature language.
- **Feature languages:** These are DSLs that used named, possibly parametrized “features” (perhaps contingent upon others) as abstractions to be realized [8].

■ **Table 4** Abstraction methods for clones in executable and declarative code

method	executable	declarative
common lambda inference (e.g., extract method) ... extended to lift lambda to common area	x	–
“template method” (abstract algorithm, interface, API) may need to merge several clones	x	–
text-based techniques	x	x
– macros, includes	x	x
– compile-time configuration (preprocessor) conditionals, file level	x	x
– frame generator (e.g., XVCL)	x	x
code generator (transformation, intention [Simony95])	x	x
runtime configuration conditionals (if/then/else, switch/case)	x	–
aspects	x	-
typedefs (e.g., structs => union)	–	x
abstract data types	–	x
object formation (e.g., legacy => OO)	x	x
normalized representation (e.g., date format)	–	x
feature formation intentional/conceptual (Type-5)?	x	?
purpose annotation for conceptual clones	x	x

4.4.7 Summary of possible code abstraction mechanisms

Most clone detection work seems to focus on cloned executable code. With CloneDR, it was observed that there are many clones in (data) declarations as well as code. We considered what kinds of abstractions might be available for code clones, and for declaration clones, to produce Table 4.

Table 4 should be extended if possible; a survey might be a useful research topic. It might be useful to collect a rather complete set of abstraction mechanisms used in software engineering (category theory, anyone?), and consider the mechanisms required to support these in clone refactoring. Which of these should be offered as a standard set to support clone refactoring opportunities? Is this standard set independent of the language in which the clones are found? How does one handle the availability of a customer-unique abstraction mechanism?

A brief discussion ensued about “clone types”. Clone Type-1 corresponds to exact-match code (perhaps modulo blanks), Type-2 to clones with single-token parameters, Type-3 to clones with larger parameters, although it is unclear how complex a parameter might be or even if it must be contiguous in the code text. Type-4 has been used to to classify clones that match semantically; since the discussion is often about clones detected with automation, presumably these are clones recognized using some semantic comparison mechanism (e.g., isomorphic dataflows), for which there are practical and theoretical limits on capability.

In considering how one might abstract code in general, it struck us that in general one might have conceptual clones, that is, blocks of code whose purpose has similar abstractable intention, but for which no mechanical detector is available (e.g., bitonic sort and radix sort routines), but have reasonable abstractions (e.g., intentions [13]). Such conceptual clones must be discovered in part by use of a human oracle. There does not seem to be a (conceptual!) problem with conceptual parameters for such clones. Should such clones have a designated type (e.g., Type-5?)

It is also a little unclear where structural clones fit in this spectrum; it is clear they have common parts including conceptual clones; as an odd extreme, one might have a structural clone that was composed entirely of conceptual subclones. Structural clones may have parameters (Type 2 or 3) induced by their component subclones, but are there “parameters” induced by the regions around the structural clones? We do not seem to have any kind of useful characterization of the nature of parameters that a clone may have; how are we to abstract without understanding what parameters might be? In the same vein, what are the parameters of variation of the structure itself?

4.4.8 Managerial Aspects of Clone Refactoring

Assuming that one can refactor clones, there is the issue of should one refactor? We briefly discussed the following:

- Decision to declone based on cost-benefit: We do not have clear economic models of the benefit of removing clones. It is becoming clearer that removing all clones may not pay off. How do we measure costs and impacts? How do we decide which ones to remove? How do we decide which abstraction mechanism to apply? How do we manage the rest, if at all?
- Cost of having changed the code: When determining cost benefit, completion of refactoring a clone is not the end of the cost; as a practical matter, changed code must be re-compiled, re-tested, re-deployed. Management often has a “do not touch anything that works” attitude partly to control this. Are some clones easier to remove? Can some be removed without doing retest? What about performance impacts of functionally reliable decloning?
- How to minimize manual refactoring work: One can automate the removal of all clones, but this is generally not a good idea. Given the possibility that each clone pair/set might be remedied differently (including not remedied), fully automated removal is likely to produce clones remedied inappropriately⁴. So there likely needs to be some interactive selection of how individual clones are removed. Given that 10% of a code based might be cloned, a million line system might have 20,000 5-line clones in 10,000 clone pairs. Interactive review of such a huge amount of data is daunting at best. Perhaps one can design defaults or heuristics so that the reviewing engineer has a simple interaction once per clone (“one click to orbit”) remediation to accept the default.

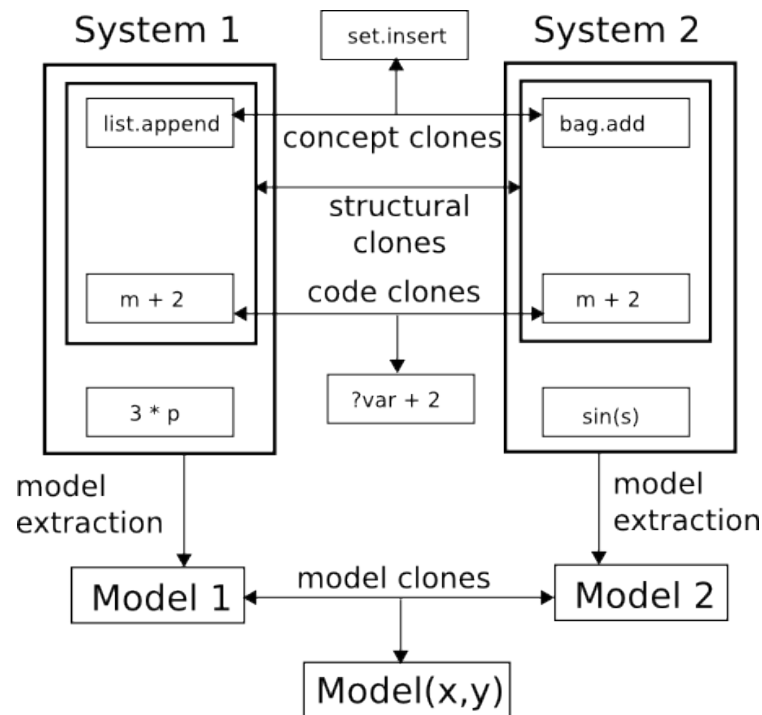
4.4.9 Refactoring To Product Lines

The code base for large software systems sometimes gets forked (multiple times!), and the resulting large-scale clones (e.g., full-code bases) then begin largely independent lives at great maintenance costs to the owning organization⁵. It is often clear after the fact that a product line should have been constructed, but the sheer scale of the systems and lack of deep understanding of process or usable tools prevent the organized construction of such a product line by somehow merging the forks. Can we refactor such enormous clones⁶ into a product line? What process and technologies are needed?

⁴ An early version of CloneDR removed all clones in a C system by converting them into macros, producing legal, compilable, runnable code, that the programmers instantly rejected because their individual code was not remedied as they would have desired.

⁵ Semantic Designs has a client with 30 copies of a large-scale core-banking system, customized to different countries legal systems and cultural product needs.

⁶ Clearly one can apply clone detection management techniques within a product instance, but for product lines, we are interested only in clones across the product line instances.



■ **Figure 3** Forming a product line from system instances

This is especially difficult in that the component languages which comprise the code base for the product line as a practical matter almost surely cannot express an abstraction that covers the entire code base. To abstract system clones, one must step outside the component languages.

What is needed are:

- means to describe the resulting product line (e.g., a specification-type of abstraction)
- methods to detect the similarities across the cloned systems
- methods to abstract the similarities into elements controlled by the specification style
- methods to manage the differences in the systems

Abstractions for classic software clones can usually be expressed in the language in which the clone was found. For product lines, often composed of multiple different computer languages as well as informal documentation, no obvious unified abstraction mechanism exists. In essence, one has to move to some kind of generator scheme in which the abstraction is expressed as some kind of specification, and a corresponding generator can produce the instance system code needed for a particular specification instance. One might choose some kind of abstract interconnection model (e.g., UML, Component/Connector architectures, [9], Module Interconnection Languages [14]) or a (set of cooperating) domain specific languages; if the latter, where does domain knowledge come into the process? A “generic” class of DSLs such as feature description languages [8] appear to be reasonable candidates for encoding the abstraction, to the extent that the features can be coupled to some kind of generative process that can produce instance systems from a selected set of feature specifications. Oversimplifying, Gears [11] suggests abstracting product lines with features that select entire files that comprise the product, and offers a commercial product for managing product lines using this technique.

Product line abstraction from code. Regardless of the final abstraction, somehow the similarities and the differences of the system clones must be found and eventually integrated into the product line. In the seminar, we generated Figure 3. We see the variety of ways in which various types of clone detection and abstraction techniques might be applied to two systems instances. At the lowest level, standard code clone detection techniques can be used to discover parametrized code abstractions that the product line generator will likely need to instantiate. Because code clone detectors cannot identify code blocks with similar intent but unsimilar code, one is likely to need to allow conceptual clones to be interactively identified; perhaps domain ontologies would be helpful. Structural clone detectors are needed to determine where sets of clones across the system instances indicate a higher level application structure; we draw attention to the fact that such structure-clone detectors must operate over the results of any of the lower-level clone detectors and even recursively over smaller detected structural clones. Any remaining code fragments not allocated to structural clones or abstracted away become (possibly enormous) parametric values of the product line instances themselves.

Product line abstraction from models. An alternative is to somehow model the systems, forming corresponding models (e.g., UML, Petri Net, ...), and apply clone detection over the models to generate an abstract model. Since feature models are models, it might be useful to build a feature model of individual systems, and do clone detection over those features. Is it possible or useful to do both abstraction from models and code in a synergistic way? We remark that structural clones might contain subclones derived from code and subclones derived from models.

Either approach leaves open the question of how the detected (structural or model) clones are abstracted back to features, specifications, or DSL elements.

It would be interesting research to manually construct a product line using clone detection processes on system instances, to provide some insight and details about how such a product line forming process might work.

4.4.10 Summary

It is remarkable how much ground one can cover in small, lively subgroup in just a few hours. The discussion was not anywhere near as linear as this report implies. This reporter has tried to do the discussion justice and augmented it somewhat, particularly adding references that seemed relevant. Any errors or misconceptions in this summary are the fault of the reporter, not the group. Thanks go to Jochen for capturing excellent notes, and to the seminar organizers for enabling us to have this discussion.

We close with a summary in Table 5 describing how to apply clone refactoring to various types of artifacts. This table should be extended to handle non-code artifacts.

4.4.11 Participants

Participants of this working group were as follows:

- Ira Baxter, Moderator, Reporter
- Sandro Schulze
- Ravindra Naik
- Hamid Abdul Basit
- Angela Lozano
- Yingnong Dang
- Jochen Quante, Scribe

■ **Table 5** Clone refactoring in various types of artifacts


<i>artifacts</i>	<i>refactoring technology</i>
code	see Table 4
data	see Table 4
feature model (UML, ontology, ...)	unknown
conceptual clones	human provided abstraction
product line instances – same language/technologies	combination of above refactorings
product line instances – different technologies	unknown. Likely conceptual clones, domain analysis/engineering

References

- 1 R. Akers, I Baxter, and M. Mehlich. Re-Engineering C++ Components Via Automatic Program Transformation. 2004 ACM SIGPLAN Symposium on Partial Evaluation and Semantics-Based Program Manipulation
- 2 Hamid Abdul Basit, Stanislaw Jarzabek. Towards Structural Clones – Analysis and Semi-Automated Detection of Design-Level Similarities in Software. VDM 2010: I-XII, 1-153
- 3 P.G. Bassett. The Case for Frame-Based Software Engineering, IEEE Software, July 2007, pp. 90–99
- 4 I.D. Baxter, C. Pidgeon, and M. Mehlich. DMS: Program Transformation for Practical Scalable Software Evolution. International Conference on Software Engineering, pp. 625–634, 2004.
- 5 I.D. Baxter, A. Yahin, L. Moura, M. Sant’Anna, and L. Bier. Clone Detection Using Abstract Syntax Trees. International Conference on Software Maintenance, IEEE Press, 1998
- 6 James Blair and Don Batory. A Comparison of Generative Approaches: XVCL and GenVoca, Department of Computer Sciences. www.cs.utexas.edu/ftp/predator/xvcl-compare.pdf
- 7 C. Brown and S. Thompson. Clone Detection and Elimination for Haskell, PEPM’10, January 18–19, 2010.
- 8 Krzysztof Czarnecki. Understanding Variability Abstraction and Realization. International Conference on Software Reuse ICSR 2011: 1-3
- 9 D. Garlan and R. Allen. Formalizing Architectural Connection, Proceedings ICSE 16, IEEE 1994.
- 10 S. Jarzabek and S.Li. Eliminating Redundancies with a ‘Composition and Adaptation’ Meta-Programming Technique, Proc. European Software Eng. Conf./ACM/SIGSOFT Symp. Foundations of Software Engineering, (ESEC/FSE 03), ACM Press, 2003, pp. 237–246;
- 11 Charles W. Krueger. The BigLever Software Gears, Systems and Software Product Line Lifecycle Framework. SPLC Workshop 2010: 297
- 12 H. Li and S. Thompson. Clone Detection and Removal for Erlang/OTP within a Refactoring Environment. ACM SIGPLAN Workshop on Partial Evaluation and Program Manipulation (PEPM’09).
- 13 Charles Simonyi. The Death of Computer Languages, the Birth of Intentional Programming (technical report) 1995
- 14 R. Prieto-Diaz and J. Neighbors. Module Interconnection Languages, Journal of Systems and Software 6, 1986.

4.5 Working group on clone management (process)

Jens Krinke (UCL, GB)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Jens Krinke

What follows are the notes kept on the seminar wiki of this working group.

4.5.1 Clones and Process

Where may clone analysis play a role in the development and maintenance process?

4.5.1.1 During Development

- As a part of QA in Continuous Integration, seen as a testing / integration / metric activity.
- It can be integrated with commits.
 - Generate commit messages automatically (“Copied X from Y.”) which can be edited and/or augmented by the user.
 - Prevent commits that would create clones (or too large clones).
 - Automatically create annotations (traceability links) between the copied code and the copy.
- It can be used before and after commits.
- During editing, providing immediate feedback (“similar code to the one you are editing exists at A, B, and C”).
- Tracking the copy/paste operations may generate useful information but may also create too much information.

4.5.1.2 During Requirements Engineering

Observation: clones in requirements may lead to semantic clones in the code (different developers implementing the same feature because of cloned requirements).

Clone detection during requirements engineering may prevent clones in later stages.

4.5.1.3 During Testing

Lots of clones exist in (unit) test suites. If code is cloned, is the test code cloned with it? Must the test code be cloned first in TDD?

4.5.1.4 After Deployment

Is my code leaking to other products? (Provenance)

4.5.2 Code Search and Cloning

Programmers use code search to find code that already does what they want to do, which is then copied. This may increase copied code – however, is this bad? Not necessarily, because cloning code is cheaper than developing a feature from scratch. Moreover, detecting of such clones is easier / possible in comparison to detect semantic clones due to reimplementations from scratch. Maybe another case of good cloning?

Other code search: search for similar / cloned code: Where does similar code exist?

Notifications of clones: what are the implications of them at edit time and commit time?

4.5.3 Recommender Systems

Recommender systems for cloned code: “You edit a clone, maybe you want to edit X and Y, too?” Implications may be similar to co-change based recommender systems: “You edit X and you have always edited X with Y and Z in the past.” However, there is no correlation between co-changes and clones. Moreover, half of the time clones evolve independently (Lague did such a study already in 1997).

- What can be recommended? (e.g., API mining)
- In which way? As wizard? As clippy?
- Depends on use / business case.

4.5.4 More questions

- What are interesting clones?
- Can clones be ranked? How?
- What to do with bugs and clones, bugs in clones?
- Are large Type-1 clones the most interesting ones?

All of the above questions may have a different answer for different development tasks. “Often, developers/management think that they are in control of the cloning and don’t have to act on it. What if they are wrong?”

- How to define “wrong”?
- Is a general question, not specific for cloning

When to

- track
 - At commit.
 - At copy/paste actions? Necessary for immediate feedback.
 - What is the granularity of the tracking?
- detect in real-time
 - Good as long as it does not get in the way.
- refactor
 - When the user needs it.
 - Depends on the use /business case.

Can copy/paste information be used for clone detection? “Maybe there is a clone...”

We need an ethnographic study.

Clone analysis may play an important role in a software product line development process.

Clones are created because of code ownership as it is hard to change other developer’s code (clone-to-own).

Larger issues: Forks are created of “social” reasons (see forks of major open-source software). We are missing an “integration culture”.

4.5.5 Participants


Participants of this working group were as follows:

- Michael Godfrey, University of Waterloo, CA
- Jindae Kim, The Hong Kong University of Science & Technology, HK
- Jens Krinke, University College London, GB

- Angela Lozano, UC Louvain-la-Neuve, BE
- Ravindra Naik, Tata Consultancy Services – Pune, IN
- Werner Teppe, Amadeus Germany GmbH, DE
- Minhaz Zibran, University of Saskatchewan, CA

4.6 Working group on provenance and clones in artifacts that are not source code

Serge Demeyer (University of Antwerp, BE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Serge Demeyer

What follows are the notes kept on the seminar wiki of this working group.

4.6.1 Clone Analysis in Binaries

4.6.1.1 Use cases

- License infringement. Example: The app store problem – is open source used in the app store?
- Malware Detection. Example: Microsoft releases a patch – detect the differences; where are you vulnerable?
- Abuse case example: detect the difference; where can I exploit?

4.6.1.2 All boils down to two different cases

1. You know that the corpus contains the subject (in that case you can try all techniques until you find whatever you are looking for).
2. You do not know whether the corpus contains the subject (in which case you can just argue adequacy; legal term = “due diligence” = I did my best to according to the state of the art in the field).

4.6.1.3 What is the information you can exploit?

- Call graphs
- Libraries used
- Signatures of methods/classes
- Strings (and constants)
- Runtime analysis (observing behaviour)
- File name
- Call usage (call graphs)
- No code (data files used, services used)
- Metrics of the binary
- Op codes
- Frequency based analysis (spectography)

Open question: what is important (in a pool of information)?

4.6.1.4 Research agenda problems/questions

- How to create traceability links to maintain the history of an artifact?
 - How to insert this into organizational process / awareness / ...?
 - How to certify the origin; what should be in the “manifest” that accompanies a software artefact?
- Diffing
 - in the case where you have two binaries which you know are descendants from one another;
 - challenging because some differences are caused by irrelevant changes (change in compiler version, options)
- Origin
 - Which point in time in the VCS was used to generate this binary? Example: You have the DEBIAN VCS and a binary; which version of DEBIAN does it come from?
 - Given a binary and the source code; is the source code the actual source code used to create the binary?
 - People copying JAR files and dropping version info; what version did I use?
 - Which version of telnet was used in malware?
- How do we evaluate that our methods are good? (\Rightarrow Benchmarks)
 - What are the common tasks?
 - Malware detection problems?
 - What in the case of obfuscation?
- Building a Corpus in the case of provenance
- Language dependent issues
- Adversarial? (incl. obfuscation)

4.6.2 Provenance

4.6.2.1 Can we automatically add some meta-data, signatures to binaries?

- Similar to EXIF for JPG files;
- currently based on a web of trust; when downloading open-source software the signatures and the binaries are kept together; there is no separate authority that authorizes signatures

4.6.2.2 Who did what post-mortem?

Manifest of a software artefact; like in ship cargo (what’s inside the container) or like a software bill of materials

- There is an industry motivated group who is standardizing this software manifest concept
 - Software Package Data Exchange
- Clarity of the supply chain; which are the organizations who produced a given component?

4.6.2.3 IEEE malware working group proposal

Working group has a taggant effort⁷. Packers compress executables. The IEEE working group would like the packer vendors to create packers that sign the packed executables with a digital signature that can be traced back to the packer vendor and packer vendor’s customer,

⁷ <http://standards.ieee.org/develop/indconn/icsg/malware.html>

and permit the signature to be revoked if the packer is stolen or the packer vendor plays both sides (sells to both white and black hats).

4.6.2.4 Where would such a “centralized authority” come from?

www.OHLOH.net: a web-site which keeps statistics of all open source software

4.6.2.5 Research Agenda

There are various non-technical issues which severely challenge the use cases applications.

Tool support is really missing. Triangulating with partial information is a potential way to go. Clone detection may contribute there.

4.6.3 Clones in models

4.6.3.1 Use Case; e.g., Simulink

Jim Cordy is looking for clones in Simulink models. GM wants to answer a question like “Is a given piece of a model –where we suspect there is a safety issue– used elsewhere in the car?” This boils down to the question we have seen elsewhere. If you discovered an issue in one model, can you look for it in others?

4.6.3.2 Observation: Culture of clones in other engineering disciplines

- In other engineering it is an accepted practice of scaling up by replicating proven designs.
- In computer science, we do not do that; we create our own abstractions and then repeat the abstractions.
- Clones are a symptom for the potential of creating such abstractions.
- However, the language must allow for “program-like” abstraction facilities.
- Most engineering disciplines lack the languages for expressing said abstractions.
- Within engineering modelling there is a new wind; with expressing higher order abstractions.
- Automotive is a good example: engineers would like to control (hence model) the emerging properties of systems.
- Example: if I push on this emergency button, will the system stop in time? The current best practice is to run many simulations and worst case scenarios.

4.6.3.3 Questions

How does duplication in models trigger abstraction?

- Having a replication of a good idea;
- pattern matching on languages/models used in other engineering disciplines is a prerequisite.

4.6.3.4 Clones in pictures

Models usually have a graphical representation; couldn’t we just use clone detection of images? Example: Getty wants to protect its copyright and spies the web for copies of its images.

- Google is now able to detect “clones” of PNG files.
- What would happen if you use that kind of facility on UML diagrams?

- Here as well: having copies of UML diagrams implies that it is a good design since people want to copy it; it is not about license infringement or so.

4.6.3.5 Research Agenda

- Look how other disciplines how they deal with duplication / replication: First thing to watch out for: Do there exist “program like” abstraction facilities?
- Reach out to other communities: Show nice examples of things we have achieved with clone research.

4.6.4 Clones in bug reports

Clones in stack traces / debug back traces (i.e., the stack traces associated with a bug).

- Clone detection there might help to identify most frequently (re-)occurring bugs.
- Canonical (the company behind the Ubuntu linux version) would be very eager for knowing which bugs occur frequently in the field.
- Integrators (= organizations combining components coming from various sources) might very interested as well. It would help them to identify which subcontractor caused the bug. Same applies for (distributed programming) teams; assigning a bug report to the right team is critical to reduce bug resolution time.

4.6.4.1 Research agenda

Clone detection on stack traces looks promising.

4.6.5 Overall research agenda

- When approaching “other” documents to search for clones there are two starting points:
 1. look for catalogs of patterns/abstractions/domain concepts; knowing what to search for is important as a first step
 2. verify whether there exist “program like” abstraction facilities in the languages used. This helps in identifying potential for removal of clones, or whether it stays at searching for similar occurrences.
- Observation: Clone detection, diffing, provenance and even search are intimately linked; a common thread throughout all what we discussed

4.6.5.1 Side note: Bizarre application for Type-4 clones

In n-version programs, verify whether the versions are indeed Type-4 clones (= semantically equivalent) but not Type-3 or lower (= syntactic equivalence).

4.6.6 Participants

Participants of this working group were as follows:

- Mike Godfrey
- Andrew Walenstein
- Serge Demeyer (scribe)
- Niko Schwarz
- Jens Krinke
- Armijn Hemel
- Daniel M. German
- Douglas Martin

Participants

- Hamid Abdul Basit
LUMS – Lahore, PK
- Ira D. Baxter
Semantic Designs – Austin, US
- Saman Bazrafshan
Universität Bremen, DE
- Michel Chilowicz
Université Paris-Est –
Marne-la-Vallée, FR
- Michael Conradt
Google – München, DE
- James R. Cordy
Queen's Univ. – Kingston, CA
- Yingnong Dang
Microsoft Research – Beijing, CN
- Serge Demeyer
University of Antwerpen, BE
- Stephan Diehl
Universität Trier, DE
- Daniel M. German
University of Victoria, CA
- Michael W. Godfrey
University of Waterloo, CA
- Nils Göde
CQSE GmbH – Garching, DE
- Jan Harder
Universität Bremen, DE
- Armijn Hemel
GPL Violations Project, NL
- Elmar Jürgens
CQSE GmbH – Garching, DE
- Cory J. Kapser
Calgary, Alberta, CA
- Jindae Kim
The Hong Kong University of
Science & Technology, HK
- Rainer Koschke
Universität Bremen, DE
- Jens Krinke
University College London, GB
- Thierry Lavoie
Ecole Polytechnique –
Montreal, CA
- Angela Lozano
UC Louvain-la-Neuve, BE
- Douglas Martin
Queen's Univ. – Kingston, CA
- Ravindra Naik
Tata Consultancy Services –
Pune, IN
- Jochen Quante
Robert Bosch GmbH –
Stuttgart, DE
- Martin P. Robillard
McGill Univ. – Montreal, CA
- Sandro Schulze
Universität Magdeburg, DE
- Niko Schwarz
Universität Bern, CH
- Werner Teppe
Amadeus Germany GmbH, DE
- Rebecca Tiarks
Universität Bremen, DE
- Gunther Vogel
Robert Bosch GmbH –
Stuttgart, DE
- Andrew Walenstein
University of Louisiana at
Lafayette, US
- Minhaz Zibran
University of Saskatchewan, CA



Information Visualization, Visual Data Mining and Machine Learning

Edited by

Daniel A. Keim¹, Fabrice Rossi², Thomas Seidl³,
Michel Verleysen⁴, and Stefan Wrobel⁵

- 1 Universität Konstanz, DE, keim@uni-konstanz.de
- 2 SAMM, Université Paris 1, FR, Fabrice.Rossi@univ-paris1.fr
- 3 RWTH Aachen, DE, seidl@informatik.rwth-aachen.de
- 4 Université Catholique de Louvain, BE, michel.verleysen@uclouvain.be
- 5 Fraunhofer IAIS – St. Augustin, DE and University of Bonn, DE, stefan.wrobel@iais.fraunhofer.de

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 12081 “Information Visualization, Visual Data Mining and Machine Learning”. The aim of the seminar was to tighten the links between the information visualisation community and the machine learning community in order to explore how each field can benefit from the other and how to go beyond current hybridization successes.

Seminar 19.–24. February, 2012 – www.dagstuhl.de/12081

1998 ACM Subject Classification H.5 Information interfaces and presentations, I.2.6 Learning, H.2.8 Database Applications, H.3.3 Information Search and Retrieval, I.5.3 Clustering

Keywords and phrases Information visualization, visual data mining, machine learning, nonlinear dimensionality reduction, exploratory data analysis

Digital Object Identifier 10.4230/DagRep.2.2.58

1 Executive Summary

Daniel A. Keim
Fabrice Rossi
Thomas Seidl
Michel Verleysen
Stefan Wrobel

License  Creative Commons BY-NC-ND 3.0 Unported license
© Daniel A. Keim, Fabrice Rossi, Thomas Seidl, Michel Verleysen, and Stefan Wrobel

Information visualization and visual data mining leverage the human visual system to provide insight and understanding of unorganized data. Visualizing data in a way that is appropriate for the user’s needs proves essential in a number of situations: getting insights about data before a further more quantitative analysis, presenting data to a user through well-chosen table, graph or other structured representations, relying on the cognitive skills of humans to show them extended information in a compact way, etc.

Machine learning enables computers to automatically discover complex patterns in data and, when examples of such patterns are available, to learn automatically from the examples how to recognize occurrences of those patterns in new data. Machine learning has proven



Except where otherwise noted, content of this report is licensed under a Creative Commons BY-NC-ND 3.0 Unported license

Information Vis., Visual Data Mining and Machine Learning, *Dagstuhl Reports*, Vol. 2, Issue 2, pp. 58–83
Editors: Daniel A. Keim, Fabrice Rossi, Thomas Seidl, Michel Verleysen, and Stefan Wrobel



Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

itself quite successful in day to day tasks such as SPAM filtering and optical character recognition.

Both research fields share a focus on data and information, and it might seem at first that the main difference between the two fields is the predominance of visual representations of the data in information visualization compared to its relatively low presence in machine learning. However, it should be noted that visual representations are used in a quite systematic way in machine learning, for instance to summarize predictive performances, i.e., whether a given system is performing well in detecting some pattern. This can be traced back to a long tradition of statistical graphics for instance. Dimensionality reduction is also a major topic in machine learning: one aims here at describing as accurately as possible some data with a small number of variables rather than with their original possibly numerous variables. Principal component analysis is the simplest and most well known example of such a method. In the extreme case where one uses only two or three variables, dimensionality reduction is a form of information visualization as the new variables can be used to directly display the original data.

The main difference between both fields is the role of the user in the data exploration and modeling. The ultimate goal of machine learning is somehow to get rid of the user: everything should be completely automated and done by a computer. While the user could still play a role by, e.g., choosing the data description or the type of algorithm to use, his/her influence should be limited to a strict minimum. In information visualization, a quite opposite point of view is put forward as visual representations are designed to be leveraged by a human to extract knowledge from the data. Patterns are discovered by the user, models are adjusted to the data under user steering, etc.

This major difference in philosophy probably explains why machine learning and information visualization communities have remained relatively disconnected. Both research fields are mature and well structured around major conferences and journals. There is also a strong tradition of Dagstuhl seminars about both topics. Yet, despite some well known success, collaboration has been scarce among researchers coming from the two fields. Some success stories are the use of state-of-the-art results from one field in the other. For instance, Kohonen's Self Organizing Map, a well known dimensionality reduction technique, has been successful partly because of its visualization capabilities which were inspired by information visualization results. In the opposite direction, information visualization techniques often use classical methods from machine learning, for instance, clustering or multidimensional scaling.

The seminar was organized in this context with the specific goal of bringing together researchers from both communities in order to tighten the loose links between them. To limit the risk of misunderstandings induced by the different backgrounds of researchers from the two communities, the seminar started with introductory talks about both domains. It was then mainly organized as a series of thematic talks with a significant portion of the time dedicated to questions and discussions. After the first two days of meeting, understanding between both communities reached a sufficient level to organize, in addition to the plenary talks, working group focusing on specific issues.

Several research topics emerged from the initial discussions and lead to the creation of the working groups. The subject that raised probably the largest number of questions and discussions is Evaluation. It is not very surprising as differences between the communities about evaluation (or quality assessment) might be considered as the concrete technical manifestation of cultural and philosophical differences between them. Indeed, in machine learning, automatic methods are mostly designed according to the following general principle: Given a quality measure for a possible solution of the problem under study, one devises an

algorithm that searches the solution space efficiently for the optimal solution with respect to this measure. For instance, in SPAM filtering a possible quality measure is the classification accuracy of the filter: it has to sort unsolicited bulk messages correctly into the SPAM class and all other emails in the HAM class. In a simple setting, the best filter could be considered as the one with the smallest number of errors. However, counting only the number of errors is usually too naive, and better quality measures have to be used, such as the area under the ROC curve: the Receiver Operating Characteristic curve shows the dependency between the true positive rate (the percentage of unsolicited bulk messages classified as SPAM) and the false positive rate (the percentage of correct emails classified as SPAM).

In information visualization, evaluation cannot rely only on mathematical quality measures as the user is always part of the story. A successful visualization is a solution, with which the user is able to perform better, in a general sense, compared to existing solutions. As in machine learning, a method is therefore evaluated according to some goal and with some quality metric, but the evaluation process and the quality metrics have to take the user into account. For instance, one display can be used to help the user assess the correlation between variables. Then, a quality metric might be the time needed to find a pair of highly correlated variables, or the time needed to decide that there is no such pair. Another metric might be the percentage of accurate decisions about the correlation of some pairs of variables. In general, a visualization system can be evaluated with respect to numerous tasks and according to various metrics. This should be done in a controlled environment and with different users, to limit the influence of interpersonal variations.

Among the discussions between members of the two communities about evaluation, questions were raised about the so-called unsupervised problems in machine learning. These problems, such as clustering or dimensionality reduction, are ill-posed in a machine learning sense: there is no unquestionable quality metric associated to e.g. clustering but rather a large number of such metrics. Some of those metrics lead to very difficult optimization problems (from a computational point of view) that are addressed via approximate heuristic solutions. In the end, machine learning has produced dozens of clustering methods and dimensionality reduction methods, and evaluations with respect to user needs remain an open problem. An important outcome of the seminar was to reposition this problem in the global picture of collaboration between information visualization and machine learning. For instance, if many quality measures are possible, one way to compare them would be to measure their link to user performances in different tasks. If several methods seem to perform equally well in a machine learning sense, then the user feedback could help to identify the «best» method. It was also noted that many methods that are studied in machine learning and linked to information visualization, in particular dimensionality reduction and embedding techniques, would benefit from more interaction between the communities. At minimum, state-of-the-art methods from machine learning should be known by information visualization researchers and state-of-the-art visualization techniques should be deployed by machine learning researchers.

Another topic discussed thoroughly at the seminar was the visualization of specific types of objects. Relational data were discussed, for instance, as a general model for heterogeneous complex data as stored in a relational database. Graph visualization techniques provide a possible starting point, but it is clear that for large databases, summarization is needed, which brought back the discussion of the ill defined clustering problem mentioned above. Among complex objects, models obtained by a machine learning algorithms were also considered, in particular as good candidates for interactive visualizations. Decision trees give a good example of such objects: Given a proper visualization of the current tree, of some possible

simplified or more complex versions and of the effect of the tree(s) on some dataset, an expert user can adapt the tree to his/her specific goals that are not directly expressible in a quality criterion. The extreme case of visualizing the dynamic evolution of a self learning process was discussed as a prototype of complex objects representation: The system is evolving through time, it learns decision rules, and it evolves using complex (and evolving) decision tables.

Finally, it became clear that a large effort is still needed at the algorithmic and software levels. First, fast machine learning techniques are needed that can be embedded in interactive visualization systems. Second, there is the need for a standard software environment that can be used in both communities. The unavailability of such a system hurts research to some extent as some active system environments in one field do not include even basic facilities from the other. One typical example is the R statistical environment with which a large part of machine learning research is conducted and whose interactive visualization capabilities are limited, in particular in comparison to the state-of-the-art static visualization possibilities. One possible solution foreseen at the seminar was the development of some dynamic data sharing standard that can be implemented in several software environments, allowing fast communication between those environments and facilitating software reuse.

Judging by the liveliness of the discussions and the number of joint research projects proposed at the end of the seminar, this meeting between the machine learning and the information visualization communities was more than needed. The flexible format of the Dagstuhl seminars is perfectly adapted to this type of meeting and the only frustration perceivable at the end of the week was that it had indeed reached its end. It was clear that researchers from the two communities were starting to understand each other and were eager to share more thoughts and actually start working on joint projects. This calls for further seminars ...

2 Table of Contents

Executive Summary

Daniel A. Keim, Fabrice Rossi, Thomas Seidl, Michel Verleysen, and Stefan Wrobel 58

Overview of Talks

Graph visualization methods and data mining: results, evaluation, and future directions	
<i>Daniel Archambault</i>	65
Steerable Large Scale Data Analytics	
<i>Daniel Archambault</i>	65
Multivariate data exploration with CheckViz and ProxiViz	
<i>Michael Aupetit</i>	65
Matrix relevance learning and visualization of labeled data sets	
<i>Michael Biehl</i>	66
Supervised dimension reduction – A brief history	
<i>Kerstin Bunte</i>	66
Overview of Visual Inference	
<i>Dianne Cook</i>	67
Eye-tracking Experiments for Visual Inference	
<i>Dianne Cook</i>	67
Future Analysis Environments	
<i>Jean-Daniel Fekete</i>	67
Psychology of Visual Analytics	
<i>Brian D. Fisher</i>	68
BCI-based Evaluation in Information Visualization	
<i>Hans Hagen</i>	68
Including prior knowledge into data visualization	
<i>Barbara Hammer</i>	69
Automated Methods in Information Visualization	
<i>Helwig Hauser</i>	69
Distance concentration and detection of meaningless distances	
<i>Ata Kaban</i>	69
Visual Analysis of Multi-faceted Scientific Data: a Survey	
<i>Johannes Kehrner</i>	70
Towards Visual Analytics	
<i>Daniel A. Keim</i>	70
Visualization of Network Centralities	
<i>Andreas Kerren</i>	71
Embedding from high- to low-dimensional spaces; how can we cope with the phenomenon of norm concentration?	
<i>John A. Lee</i>	71

Visual Analytics of Sparse Data	
<i>Marcus A. Magnor</i>	72
Exploration through Enrichment	
<i>Florian Mansmann</i>	72
Quality metrics for InfoVis	
<i>Florian Mansmann</i>	72
The Generative Topographic Mapping and Interactive Visualization	
<i>Ian Nabney</i>	73
Information Retrieval Perspective to Nonlinear Dimensionality Reduction for Data Visualization	
<i>Jaakko Peltonen</i>	73
Visualization of Learning Processes – A Problem Statement	
<i>Gabriele Peters</i>	74
Learning of short time series	
<i>Frank-Michael Schleif</i>	74
Comparative Visual Cluster Analysis	
<i>Tobias Schreck</i>	75
Visualization of (machine) learning processes and dynamic scenarios	
<i>Marc Strickert</i>	75
Prior knowledge for Visualization or prior visualization results for knowledge generation – a chicken-egg problem?	
<i>Holger Theisel</i>	76
Interactive decision trees and myriahedral maps	
<i>Jarke J. Van Wijk</i>	76
Clustered graph, visualization, hierarchical visualization	
<i>Nathalie Villa-Vialaneix</i>	76
Perceptual Experiments for Visualization	
<i>Daniel Weiskopf</i>	77
Introduction to embedding	
<i>Laurens van der Maaten</i>	77

Working Groups





Results of Working Group: Visualization of Dynamic Learning Processes	
<i>Michael Biehl, Kerstin Bunte, Gabriele Peters, Marc Strickert, and Thomas Villmann</i>	78
Results of Working Group: Model Visualization – Towards a Tight Integration of Machine Learning and Visualization	
<i>Florian Mansmann, Tobias Schreck, Etienne Come, and Jarke J. Van Wijk</i>	78
Results of Working Group: Embedding techniques at the crossing of Machine Learning and Information Visualization	
<i>Michael Aupetit and John Lee</i>	79

Results of Working Group: Evaluation <i>Ian Nabney, Dianne Cook, Brian D. Fisher, Andrej Gisbrecht, Hans Hagen, and Heike Hoffman</i>	80
Results of Working Group: Fast Machine Learning <i>Jörn Kohlhammer</i>	80
Results of Working Group: Future analysis environments <i>Jean-Daniel Fekete</i>	81
Results of Working Group: Structured/relational data <i>Nathalie Villa-Vialaneix</i>	81
Participants	83

3 Overview of Talks

3.1 Graph visualization methods and data mining: results, evaluation, and future directions

Daniel Archambault (University College Dublin, IE)





License     Creative Commons BY-NC-ND 3.0 Unported license
© Daniel Archambault

Graph visualization and data mining methods have many areas of common interest.

In the introductory talk for this session, I will cover some of my recent results on graph visualization applicable to this topic, outline methods of visualization research, and identify some possible areas of future collaboration.

3.2 Steerable Large Scale Data Analytics





Daniel Archambault (University College Dublin, IE)

License     Creative Commons BY-NC-ND 3.0 Unported license
© Daniel Archambault

In this short talk, I cover some ideas on steerable data analytics. In this area, I think that we should strive to strengthen the coupling between data mining or clustering processes and visualization in order to enable real time analysis. I give potential ways to achieve this goal with possible applications to the area of social media analysis and community finding.

3.3 Multivariate data exploration with CheckViz and ProxiViz

Michael Aupetit (Commissariat à l'Energie Atomique – Gif-sur-Yvette, FR)

License     Creative Commons BY-NC-ND 3.0 Unported license
© Michael Aupetit

Joint work of Aupetit, Michael; Lespinats, Sylvain

Main reference S. Lespinats, M. Aupetit, “CheckViz: Sanity Check and Topological Clues for Linear and Non-Linear Mappings,” *Computer Graphics Forum* 30(1):113–125, 2011.

URL <http://dx.doi.org/10.1111/j.1467-8659.2010.01835.x>

Embedding techniques are used for multivariate data analysis. They provide a planar set of points whose relative distances estimates the original similarities.


We argue that this set of points alone is not enough to make sense out of it. We present CheckViz [2] and ProxiViz [1] as two ways to make the set of points interpretable by the user. CheckViz overload distortions straight into the map, it can be used as a sanity check and also provides inference rule which help to recover the original data topology. ProxiViz overload the true original similarity measure between a selected point and each of the other points which makes possible to reconstruct the original data structure. The embeddings appear not to be an end, but just a mean to display a complementary information which make them usable and useful for multivariate data exploration.

References

- 1 Michaël Aupetit, *Visualizing distortions and recovering topology in continuous projection techniques*. *Neurocomputing* 70(7-9):1304-1330, March 2007.
- 2 Sylvain Lespinats, and Michaël Aupetit, *CheckViz: Sanity Check and Topological Clues for Linear and Non-Linear Mappings*. *Computer Graphics Forum* 30(1):113-125, 2011.

3.4 Matrix relevance learning and visualization of labeled data sets

Michael Biehl (University of Groningen, NL)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Michael Biehl

Joint work of Biehl, Michael; Bunte, Kerstin; Hammer, Barbara; Schneider, Petra; Villmann, Thomas

A brief introduction is given to Learning Vector Quantization (LVQ) as an intuitive, flexible, and very powerful prototype-based classifier.

The focus is on the recent extension of LVQ by Matrix Relevance Learning. In this scheme, one or several matrices of adaptive relevances are employed to parameterize a distance measure.

Matrix Relevance Learning makes use of a low-dimensional linear or locally linear representation of the data set, internally. This fact can be exploited for the discriminative visualization of labelled data sets.

In terms of a few application examples from the life sciences it is argued that these visualizations facilitate valuable insight into the nature of the problems.

Possible routes to extend the schemes to explicitly non-linear visualizations are briefly discussed. This leads to the question what the goal of visualizing labeled data should be.


The following references may serve as a starting point to get acquainted with Matrix Relevance Learning in the context of visualization.

References

- 1 P. Schneider, M. Biehl, and B. Hammer. *Adaptive relevance matrices in Learning Vector Quantization*. Neural Computation 21(12): 3532-3561, 2009
- 2 K. Bunte, P. Schneider, B. Hammer, F.-M. Schleif, T. Villmann, and M. Biehl. *Limited rank matrix learning, discriminative dimension reduction, and visualization*. Neural Networks 26: 159-173, 2012

3.5 Supervised dimension reduction – A brief history

Kerstin Bunte (Universität Bielefeld, DE)


License  Creative Commons BY-NC-ND 3.0 Unported license
© Kerstin Bunte

Joint work of Bunte, Kerstin; Biehl, Michael; Hammer, Barbara

Due to improved sensor technology, dedicated data formats and rapidly increasing digitalization capabilities the amount of electronic data increases dramatically since decades. As a consequence the manual inspection data sets often becomes infeasible. In recent years, many powerful non-linear dimension reduction techniques have been developed which provide a visualization of complex data sets. Using prior knowledge, e.g. in form of supervision might provide more informative mappings dependent on the actual data set.

3.6 Overview of Visual Inference

Dianne Cook (Iowa State University, US)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Dianne Cook


Main reference A. Buja, D. Cook, H. Hofmann, M. Lawrence, E.-K. Lee, D.F. Swayne, H. Wickham, "Statistical Inference for Exploratory Data Analysis and Model Diagnostics," Royal Society Philosophical Transactions A, vol. 367, no. 1906, pp. 4361–4383, 2009.

URL <http://dx.doi.org/10.1098/rsta.2009.0120>

Implicitly detection of patterns in a plot of data is a rejection of some null hypothesis. What patterns might we see in the plot if the data was sampled in a manner consistent with the null hypothesis? This research area provides methods for assessing whether what we see in plots is "real", and obtaining levels of significance for findings based on visualization. Two protocols are used, a lineup and a rorschach. In the lineup, the plot of real data is embedded in a field of plots of data generated in a manner consistent with the relevant null hypothesis. In a rorschach, all plots are null plots, and the approach is a way to examine how much variability can occur purely by chance.

3.7 Eye-tracking Experiments for Visual Inference


Dianne Cook (Iowa State University, US)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Dianne Cook

Visual inference provides methods for assessing whether what we see in plots is real. The primary method is a lineup, where a plot of the actual data is embedded in a field of plots of data generated in a manner consistent with the null hypothesis. For example, to assess the relationship between two variables with a scatterplot, the null plots may show the same data, with one of the variables having its values permuted, thus breaking any real association between the variables. If the observer picks the actual data plot from the lineup it lends significance to the conclusion of a real relationship between the two variables. Following a series of Amazon Turk experiments where the lineup protocol was evaluated under controlled simulated data experimental conditions, we selected a handful of lineups for detailed assessment. Here subjects were recorded with an eye trackers to examine (1) how long they looked at their selection, (2) which plots caught the subjects attention and (3) how subjects scanned the lineups to make their selections.

3.8 Future Analysis Environments


Jean-Daniel Fekete (Université Paris Sud – Orsay, FR)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Jean-Daniel Fekete

What is the future of Analysis environments allowing machine learning and visualization to interoperate seamlessly? Should we design a new system that will solve all the problems, reuse already existing systems or in-between? These slides summarize a possible way to address the issue that might address the problem is a simple enough way.

3.9 Psychology of Visual Analytics

Brian D. Fisher (Simon Fraser University – Surrey, CA)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Brian D. Fisher

Main reference B. Fisher, T.M. Green, R. Arias-Hernández, “Visual Analytics as a Translational Cognitive Science,” *Topics in Cognitive Science* 3,3 609–625, 2011.


URL <http://dx.doi.org/10.1111/j.1756-8765.2011.01148.x>

This talk explores the larger implications of visual analytics – the science of analytical reasoning facilitated by interactive visual interfaces for cognitive science and informatics. The visual analytics approach emphasizes the design of technologies to support the ability of trained human analysts to understand situations, make decisions, generate plans, and put them into action. The resulting visual information systems succeed when they enable analysts to more effectively work with complex "big data" from sensors, archives, computational and mathematical models, alone and in collaboration with other analysts.

My laboratory begins by building field study methods that characterize human and computational cognitive capabilities as they are used for decision-making in specific situations in flight safety, public health, and emergency management analysis. These field studies generate research questions and experimental protocols that are used to investigate human-computer cognitive systems in the laboratory. My talk will briefly discuss our "pair analytics" methods derived from H. Clark's Joint Activity Theory and two laboratory studies of perceptual cognition in display environments similar to those proposed for air traffic control and collaborative aircraft CAD.

3.10 BCI-based Evaluation in Information Visualization

Hans Hagen (TU Kaiserslautern, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Hans Hagen

Joint work of Hagen, Hans; Ebert, Achim; Cernea, Daniel

Main reference D. Cernea, P.-S. Olech, A. Ebert, A. Kerren, “Measuring Subjectivity – Supporting Evaluations with the Emotiv EPOC Neuroheadset,” *Künstliche Intelligenz/KI Journal*, Special Issue on Human-Computer Interaction, Volume 26, Number 2 (2012), 177–182.

URL <http://dx.doi.org/10.1007/s13218-011-0165-0>


Evaluations have been the key factor for validating different visualization and interaction approaches. But while experts agree on their importance, the evaluation techniques currently used in Information Visualization focus mostly on objective measurements like performance and efficiency, and only rarely investigate subjective factors (states of mind and emotions that the users experience).

As the ideal evaluation should be non-intrusive and executed in real-time, many researchers turn to novel brain-computer interfaces (BCI) for directly investigating the users' affective and mental states. While current portable BCI systems are employed overwhelmingly in control tasks (e.g. moving a robotic Arm), many of them have proven useful in supporting subjectivity measurements and, thus, evaluations in real-time.

But what would an ideal BCI system detect and how would it process it in order to support the evaluation of Information Visualization approaches? Could a framework specifically designed for InfoVis evaluation with BCI systems enable researchers to obtain the answers they seek? These are a couple of specific topics that need to be addressed when looking at the potential of BCI systems as an alternative evaluation method for Information Visualization techniques and systems.

3.11 Including prior knowledge into data visualization

Barbara Hammer (*Universität Bielefeld, DE*)


License  Creative Commons BY-NC-ND 3.0 Unported license
© Barbara Hammer

In this presentation, the question of how data visualization and dimensionality reduction are linked to prior knowledge will be investigated. First, it will be motivated, that visualization and prior knowledge are closely connected.

Afterwards, technical possibilities how to integrate different kind of prior knowledge into dimensionality reduction will be discussed. Four major principles will be identified and demonstrated by examples: (i) change of the prior in a Bayesian model. For a cost-function based techniques, possibilities are given by a (ii) change of the data representation or metric, (iii) change of the cost function used for training, (iv) change or bias of the mapping of data to low dimensions.

3.12 Automated Methods in Information Visualization


Helwig Hauser (*University of Bergen, NO*)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Helwig Hauser

Visualization and Machine Learning have related goals in terms of helping analysts to understand characteristic aspects of data. While visualization aims at involving the user through interactive depictions of data, machine learning is generally represented by automatic methods that yield optimal results with respect to certain initially specified tasks. Not at the least within the research direction of visual analytics it seems promising to think about opportunities to integrate both methodologies in order to exploit the strengths of both sides. Up to now, examples of integration very often encompass the visualization of results from automatic methods as well as attempts to make originally automated methods partially interactive. A vision for the future would be to integrate interactive and automatic methods in order to solve problems. A possible realization could be an iterative process where the one or other approach is chosen on demand at each step.

3.13 Distance concentration and detection of meaningless distances

Ata Kaban (*University of Birmingham, GB*)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Ata Kaban

Main reference A Kaban, “Non-parametric Detection of Meaningless Distances in High-Dimensional Data,” *Statistics and Computing*. 22(1): 375–385.

URL <http://dx.doi.org/10.1007/s11222-011-9229-0>

Distance concentration is a counter-intuitive aspect of the curse of dimensionality, the phenomenon that in certain conditions the contrast between the nearest and the farthest neighbouring points vanishes as the data dimension increases. This makes distances meaningless, exponentially slows down data retrieval, and risks to compromise our ability to extract meaningful information from high dimensional data sets. First, we show that the known


sufficient conditions are also the necessary conditions of distance concentration in the limit of infinite dimensions. We then quantify the phenomenon more precisely, for possibly high but finite dimensional settings in a distribution-free manner, by bounding the tails of the probability that distances become meaningless. We show how this can be turned into a statistical test to assess the concentration of a given distance function in some unknown data distribution solely on the basis of an available data sample from it. This can be used to test and detect problematic cases more rigorously than it has been possible previously, and we demonstrate the working of this approach on both synthetic data and ten real-world data sets from different domains.

References

- 1 A Kaban. *Non-parametric Detection of Meaningless Distances in High-Dimensional Data*. Statistics and Computing. 22(1): 375-385.

3.14 Visual Analysis of Multi-faceted Scientific Data: a Survey

Johannes Kehrler (VRVis – Wien, AT)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Johannes Kehrler

Joint work of Kehrler, Johannes; Hauser, Helwig


Main reference J. Kehrler, H. Hauser, “Visualization and Visual Analysis of Multi-faceted Scientific Data: A Survey,” *IEEE Trans. on Visualization and Computer Graphics*, *accepted for publication*.

URL <http://dx.doi.org/10.1109/TVCG.2012.110>

Interactive visual analysis plays an important role in studying different kinds of scientific data (e.g., spatial, temporal and/or multi-variate data). The talk is based on a thorough literature review, which investigates to which degree methods for 1) visual representation, 2) user interaction and 3) computational analysis are combined in such an analysis. A task-based categorization of approaches is proposed and different options for the visual analysis are discussed. This leads to conclusions with respect to promising research directions, for instance, to pursue new solutions that combine supervised machine learning with interactive feature specification via brushing.

3.15 Towards Visual Analytics

Daniel A. Keim (Universität Konstanz, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Daniel A. Keim

Joint work of Keim, Daniel A.; Kohlhammer, Jörn; Geoffrey, Ellis; Mansmann, Florian

Main reference D. Keim, J. Kohlhammer, G. Ellis, F. Mansmann, (eds.), “Mastering the Information Age – Solving Problems with Visual Analytics,” Eurographics, 2010.

URL <http://www.vismaster.eu/book/>

Many of the grand challenges require not only automatic methods, but also exploration to find appropriate solutions. Visual Analytics as the tight integration of visual and automatic data analysis methods for information exploration and scalable decision support aims at integrating machine capabilities (e.g., data storage, numerical computation or search) with human capabilities (such as perception, creativity and general knowledge). Besides giving an introduction to Visual Analytics and information visualization, this talk describes common evaluation approaches and outlines the relation between visualization and machine learning.

3.16 Visualization of Network Centralities

Andreas Kerren (*Linnaeus University – Växjö, SE*)

License © © © Creative Commons BY-NC-ND 3.0 Unported license
© Andreas Kerren

Joint work of Kerren, Andreas; Köstinger, Harald; Zimmer, Björn

Main reference A. Kerren, H. Köstinger, B. Zimmer, “ViNCent – Visualization of Network Centralities,” in Proc. of the Int’l Conf. on Information Visualization Theory and Applications (IVAPP ’12), pp. 703–712, Rome, Italy, 2012. INSTICC.

The use of network centralities in the field of network analysis plays an important role when the relative importance of nodes within the network topology should be rated. A single network can easily be represented by the use of standard graph drawing algorithms, but not only the exploration of one centrality might be important: the comparison of two or more of them is often crucial for a better understanding. When visualizing the comparison of several network centralities, we are facing new problems of how to show them in a meaningful way. For instance, we want to be able to track all the changes of centralities in the networks as well as to display the single networks as best as possible. In the life sciences, centrality measures help scientists to understand the underlying biological processes and have been successfully applied to different biological networks. The aim of this talk was to briefly present a system for the interactive visualization of biochemical networks and its centralities. Researchers can focus on the exploration of the centrality values including the network structure without dealing with visual clutter or occlusions of nodes. Simultaneously, filtering based on statistical data concerning the network elements and centrality values supports this.

3.17 Embedding from high- to low-dimensional spaces; how can we cope with the phenomenon of norm concentration?

John A. Lee (*Université Catholique de Louvain, BE*)

License © © © Creative Commons BY-NC-ND 3.0 Unported license
© John A. Lee

Joint work of Lee, John A.; Verleysen, M.

Main reference J.A. Lee, M. Verleysen, “Shift-invariant similarities circumvent distance concentration in stochastic neighbor embedding and variants,” *Procedia Computer Science* Volume 4, 2011, Pages 538–547.


URL <http://dx.doi.org/10.1016/j.procs.2011.04.056>

Dimensionality reduction aims at representing high-dimensional data in low-dimensional spaces, mainly for visualization and exploratory purposes. As an alternative to projections on linear subspaces, nonlinear dimensionality reduction, also known as manifold learning, can provide data representations that preserve structural properties such as pairwise distances or local neighborhoods. Very recently, similarity preservation emerged as a new paradigm for dimensionality reduction, with methods such as stochastic neighbor embedding and its variants. Experimentally, these methods significantly outperform the more classical methods based on distance or transformed distance preservation.

This talk explains both theoretically and experimentally the reasons for these performances. In particular, it details (i) why the phenomenon of distance concentration is an impediment towards efficient dimensionality reduction and (ii) how SNE and its variants circumvent this difficulty by using similarities that are invariant to shifts with respect to squared distances. The paper also proposes a generalized definition of shift-invariant similarities that extend the applicability of SNE to noisy data.

3.18 Visual Analytics of Sparse Data


Marcus A. Magnor (TU Braunschweig, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Marcus A. Magnor

High-dimensional data will always constitute only sparse representations of inter-dimensional information. As a result of large voids in n-D space, even without taking noise and erroneous data into account, putative inter-dimensional relations may only be hallucinated, by humans as well as by algorithms. In contrast, suitable interpolation on the data level, guided by high-level knowledge of the data and dimensional meaning, may be able to plausibly fill the voids and to fortify subsequent interactive and automatic analysis results.

3.19 Exploration through Enrichment

Florian Mansmann (Universität Konstanz, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Florian Mansmann

Joint work of Mansmann, Florian; Spretke, David; Janetzko, Halldor; Bak, Peter
Main reference D. Spretke, H. Janetzko, F. Mansmann, P. Bak, B. Kranstauber, S. Davidson, M. Mueller, “Exploration through Enrichment: A Visual Analytics Approach for Animal Movement,” in Proc. of the 19th SIGSPATIAL Int’l Conf. on Advances in Geographic Information Systems, pp. 421–424, 2011.
URL <http://dx.doi.org/10.1145/2093973.2094038>


In many visualization scenarios, visualizing and exploring data raises hypotheses that cannot be answered with the current data. Therefore, very often an enrichment phase is needed to enhance the exploration process. In this talk, I showed two prototypes, namely ClockView in which network time series can be filtered through user-defined patterns and the Animal Ecology Explorer in which bird movement can be interactively refined through machine learning methods such as clustering and classification.

References

- 1 C. Kintzel, J. Fuchs and F. Mansmann. *Monitoring Large IP Spaces with ClockView*. Proc. of Int. Symp. on Visualization for Cyber Security (VizSec), 2011.
- 2 D. Spretke, H. Janetzko, F. Mansmann, P. Bak, B. Kranstauber, S. Davidson and M. Mueller. *Exploration through Enrichment: A Visual Analytics Approach for Animal Movement*. Proc. of the 19th SIGSPATIAL International Conference on Advances in Geographic Information Systems, pages 421–424, 2011.

3.20 Quality metrics for InfoVis

Florian Mansmann (Universität Konstanz, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Florian Mansmann

Joint work of Bertini, Enrico; Tatu, Andrada; Keim, Daniel A.
Main reference E. Bertini, A. Tatu, D.A. Keim, “Quality Metrics in High-Dimensional Data Visualization: An Overview and Systematization,” IEEE Symposium on Information Visualization (InfoVis), 2011.
URL <http://dx.doi.org/10.1109/TVCG.2011.229>

Quality metrics are a recent trend in the information visualization community. The basic idea is that the quality of a visualization with respect to the loaded data can be calculated


and based on this assessment the good or optimal parameter configurations for visualizations can be found.

References

- 1 E. Bertini, A. Tatu and D. A. Keim. *Quality Metrics in High-Dimensional Data Visualization: An Overview and Systematization*. IEEE Transactions on Visualization and Computer Graphics (TVCG), vol. 17, no. 12, pp. 2203-2212, 2011.

3.21 The Generative Topographic Mapping and Interactive Visualization

Ian Nabney (Aston University – Birmingham, GB)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Ian Nabney


Joint work of Nabney, Ian; Tino, Peter; Maniyar, Dharmesh; Schroeder, Martin

The Generative Topographic Mapping (GTM) is a probabilistic generative data model. Using Bayes' theorem, the mapping can be inverted and used for visualization. Because the model is a constrained mixture of Gaussians, an (extended) EM algorithm can be used to train models. The smooth mapping defined by GTM defines a two-dimensional manifold embedded in data space: geometric measures (e.g. magnification and curvature) can be visualized to understand the embedding and diagnose modelling flaws.

More recent advances include modelling missing data, discrete variables, and hierarchies: all of these can be handled in a consistent probabilistic framework. With a bit more analysis, it is possible to incorporate prior knowledge of variable correlation structure (with block-structured covariance models) and unsupervised feature selection (with minimum message length criteria). The talk concluded with a short demonstration of a visualization system that integrates machine learning and information visualisation (Data Visualization and Modelling System: DVMS) written in Matlab which is available from the Aston website. <http://www1.aston.ac.uk/eas/research/groups/ncrg/resources/netlab/downloads/>

3.22 Information Retrieval Perspective to Nonlinear Dimensionality Reduction for Data Visualization

Jaakko Peltonen (Aalto University, FI)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Jaakko Peltonen

Joint work of Venna, Jarkko; Peltonen, Jaakko; Nybo, Kristian; Aidos, Helena; Kaski, Samuel

Main reference J. Venna, J. Peltonen, K. Nybo, H. Aidos, S. Kaski, "Information Retrieval Perspective to Nonlinear Dimensionality Reduction for Data Visualization," *Journal of Machine Learning Research*, 11:451–490, 2010.

URL <http://jmlr.csail.mit.edu/papers/v11/venna10a.html>

Nonlinear dimensionality reduction methods are often used to visualize high-dimensional data, although the existing methods have been designed for other related tasks such as manifold learning. It has been difficult to assess the quality of visualizations since the task has not been well-defined. We give a rigorous definition for a specific visualization task, resulting in quantifiable goodness measures and new visualization methods. The task is information retrieval given the visualization: to find similar data based on the similarities

shown on the display. The fundamental tradeoff between precision and recall of information retrieval can then be quantified in visualizations as well. The user needs to give the relative cost of missing similar points vs. retrieving dissimilar points, after which the total cost can be measured. We then introduce a new method NeRV (neighbor retrieval visualizer) which produces an optimal visualization by minimizing the cost. We further derive a variant for supervised visualization; class information is taken rigorously into account when computing the similarity relationships. We show empirically that the unsupervised version outperforms existing unsupervised dimensionality reduction methods in the visualization task, and the supervised version outperforms existing supervised methods.

3.23 Visualization of Learning Processes – A Problem Statement

Gabriele Peters (FernUniversität in Hagen, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Gabriele Peters

The visual representation of the results of machine learning algorithms can be regarded as an open research topic. But rather to restrict the visualization to only the results of machine learning approaches, the discussion should be expanded to the visualization of the learning processes themselves. Whereas a visualization of results promises a better interpretation of what has been learned, the visualization of learning processes may provide a better understanding of underlying principles of learning (also in biological systems).

Maybe it can also account for general insights in the possibilities of autonomous learning at all. In my talk I present briefly the architecture of a self-learning system with two levels of hierarchy together with some results obtained in a computer vision task. From this I derive questions of general interest such as possible options to visualize the flow of information in a dynamic learning system or the visualization of symbolic data.

3.24 Learning of short time series

Frank-Michael Schleif (Universität Bielefeld, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Frank-Michael Schleif

Joint work of Schleif, Frank-Michael; Gisbrecht, Andrej; Hammer, Barbara
Main reference F.-M. Schleif, A. Gisbrecht, B. Hammer, “Relevance learning for short high-dimensional time series in the life sciences,” IJCNN’12, 2012

The talk presented some concepts used to learn short and high dimensional time series.

Especially I detailed a method for topographic mapping and recent extensions thereof in the line of supervised relevance learning.

Challenges in the modeling and visualization were discussed.

3.25 Comparative Visual Cluster Analysis

Tobias Schreck (Universität Konstanz, DE)

License © © © Creative Commons BY-NC-ND 3.0 Unported license
© Tobias Schreck

Joint work of Schreck, Tobias; Tatu, Andrada; Maaß, Fabian; Bertini, Enrico; Keim, Daniel

Data that is to be analyzed with cluster analysis tools may be represented by sets of feature vectors stemming from alternative feature extraction processes.

Interesting cluster structures may reside in several of the alternative feature representations, and they may confirm, complement, or contradict each other. In this talk we consider the problem of comparative visual cluster analysis in multiple features spaces (or subspaces). We first briefly review a previously proposed method for visual comparison of multiple feature spaces represented by Self-Organizing Map models. We then discuss ongoing work that aims to make use of automatic subspace selection methods. First results based using the SURFING subspace selection method are reported. The basic idea is to define a custom similarity function for the subspaces. The function currently considers the intersection of the selected dimensions as well as the agreement in clustering structures exhibited in the subspaces. Different visual representations based on MDS layouts, TreeMap layouts etc. as well as interaction techniques are investigated. Eventually, our approach should help analysts in identifying the most interesting subspaces from a potentially much larger set of subspaces reported by the subspace selection method.

3.26 Visualization of (machine) learning processes and dynamic scenarios

Marc Strickert (Universität Marburg, DE)

License © © © Creative Commons BY-NC-ND 3.0 Unported license
© Marc Strickert


The title can be related to an overwhelming plenitude of aspects such as functional brain imaging, motion sensor and eye tracker analysis, neural spike train observations, phase space portraits, or time series and data stream mining. To focus the wide topic on one essential commonality, this involves the transformation of spatio-temporal multi-dimensional input data into representations that are compatible with analysts' world view. This requires a compatibility between the data model and the world model mainly constituted by three spatial coordinates, color, intensity, and experience of spatio-temporal contiguity.

In machine learning methods sequential signals are often recursively mixed with a representation of the most recent internal state for modeling first-order context. The current model state is thus a representation of possibly unifying encodings of external dynamics. Depending on different readout functions applied to the model parameters different aspects of the input stream are focused on.

After all, structure detection, including ordering and convergence trends, is considered as crucial component for extracting aspects which are potentially relevant for visualization.

3.27 Prior knowledge for Visualization or prior visualization results for knowledge generation – a chicken-egg problem?


Holger Theisel (Universität Magdeburg, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Holger Theisel

It seems that both communities – Machine Learning and Visualization – use different words for the same concept, and even use the same words for different concepts. This holds both for “prior knowledge” and “visualization”. Being aware of this, the following questions are discussed: Are there new ML algorithms when the goal is preparation for a interactive visual analysis? Are there new Vis approaches when the goal is not complete insight but preparation of an automatic analysis?

3.28 Interactive decision trees and myriahedral maps

Jarke J. Van Wijk (TU Eindhoven, NL)


License  Creative Commons BY-NC-ND 3.0 Unported license
© Jarke J. Van Wijk

Joint work of Van Wijk, Jarke J.; van den Elzen, Stef;
Main reference S. van den Elzen, J.J. van Wijk, “BaobabView: Interactive construction and analysis of decision trees,” IEEE VAST 2011: 151–160.

In my talk, I present BaobabView, a system developed by Stef van den Elzen. It enables users to construct, inspect, and evaluate decision trees via a wide range of features. Next, myriahedral projections are presented via a video. These are mappings of the sphere to the plane using an approximation of the sphere with a large number of facets, which are cut and folded out. Finally, a short demo of SeifertView is given. Seifert surfaces are orientable surfaces that are bounded by knots or links. They illustrate that 2-manifolds embedded in 3D can take complex shapes.

3.29 Clustered graph, visualization, hierarchical visualization

Nathalie Villa-Vialaneix (Université Paris I, FR)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Nathalie Villa-Vialaneix

Joint work of Villa-Vialaneix, Nathalie; Rossi, Fabrice
Main reference F. Rossi, N. Villa-Vialaneix, “Représentation d’un grand réseau à partir d’une classification hiérarchique de ses sommets,” Journal de la Société Française de Statistique, 152, pp. 34–65, 2011.
URL <http://hal.archives-ouvertes.fr/hal-00651577/>

Clustering is a useful approach to provide a simplified and meaningful representation of large graphs. By extracting dense communities of nodes, the “big picture” of the network organization is enlightened. Moreover, hierarchical clustering may help the user to focus on some parts of the graph which is of interest for him and which can be displayed with finer and finer details.


This talk will try to present some open issues with graph visualization based on a hierarchical nodes clustering. These issues include displaying the clusters in a coherent way between the different layers of the hierarchy or integrating information about the clustering evaluation in the visualization. It is related to the article [1].

References

- 1 Rossi, F. and Villa-Vialaneix, N. (2011) Représentation d'un grand réseau à partir d'une classification hiérarchique de ses sommets. *Journal de la Société Française de Statistique*, 152, 34-65.

3.30 Perceptual Experiments for Visualization

Daniel Weiskopf (Universität Stuttgart, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Daniel Weiskopf

I briefly describe and discuss a few examples of user experiments that investigate the visual perception of visualization results, including studies that use methods from vision research, eye-tracking experiments in the context of the visualization of node-link diagrams of graphs and trees, as well as learning attention models for video visualization by utilizing eye-tracking data.

References

- 1 Michael Burch, Corinna Vehlow, Natalia Konevtsova, Daniel Weiskopf: Evaluating Partially Drawn Links for Directed Graph Edges. *Graph Drawing 2011*, 226-237 (2011)
- 2 Michael Burch, Corinna Vehlow, Fabian Beck, Stephan Diehl, Daniel Weiskopf: Parallel Edge Splatting for Scalable Dynamic Graph Visualization. *IEEE Trans. Vis. Comput. Graph.* 17(12): 2344-2353 (2011)
- 3 Benjamin Höferlin, Hermann Pflüger, Markus Höferlin, Gunther Heidemann, Daniel Weiskopf: Learning a Visual Attention Model for Adaptive Fast-Forward in Video Surveillance. *Proceedings of International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, 25-32 (2012).

3.31 Introduction to embedding

Laurens van der Maaten (TU Delft, NL)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Laurens van der Maaten

In this talk, I presented an overview of some major embedding techniques and explained their strengths and weaknesses. In particular, I explained principal components analysis, locally linear embedding, and t-distributed stochastic neighbor embedding. In addition, I showed examples of embedding techniques that go beyond traditional dimensionality reduction and multidimensional scaling.

In particular, I covered embedding techniques that learn representations from non-metric similarities such as word associations, co-occurrences, and partial order rankings.

4 Working Groups

4.1 Results of Working Group: Visualization of Dynamic Learning Processes

Michael Biehl, Kerstin Bunte, Gabriele Peters, Marc Strickert, and Thomas Villmann

License © © © Creative Commons BY-NC-ND 3.0 Unported license
© Michael Biehl, Kerstin Bunte, Gabriele Peters, Marc Strickert,
and Thomas Villmann

Main reference T. Leopold, G. Kern-Isberner, G. Peters, "Combining Reinforcement Learning and Belief Revision – A Learning System for Active Vision," 19th British Machine Vision Conference (BMVC 2008), edited by M. Everingham, Ch. Needham, and R. Fraile, Vol. 1, pp. 473-482, Leeds, UK, 2008.

We took the learning system [1] proposed by G. Peters in her talk "Visualization of Learning Processes - A Problem Statement" as example for a dynamic learning process and figured out which components can be visualized and by which means. The system has two learning levels: one with if-then rules (boolean expressions) and one with qualities of state- action pairs. Relevant questions to ask are: What are parameters of the system? Which parameters define states to be visualized? How to visualize the list of rules and the state- action table? How to visualize dynamic processes between the hierarchy levels? We proposed several means and solutions to answer these questions and intend to submit these considerations as position paper to a suitable conference or workshop.

References

- 1 T. Leopold and G. Kern-Isberner and G. Peters. *Combining Reinforcement Learning and Belief Revision – A Learning System for Active Vision*. BMVC, 2008

4.2 Results of Working Group: Model Visualization – Towards a Tight Integration of Machine Learning and Visualization

Florian Mansmann, Tobias Schreck, Etienne Come, and Jarke J. Van Wijk

License © © © Creative Commons BY-NC-ND 3.0 Unported license
© Florian Mansmann, Tobias Schreck, Etienne Come, and Jarke J. Van Wijk

Choosing and configuring an appropriate Machine Learning model to solve a given analysis task is crucial for arriving at useful results. Models in Machine Learning are potentially complex and sometimes hard to understand for non-experts, and often regarded and applied as black boxes. In this working group we discussed about approaches to 'opening' the black boxes by visualizing not only the data space, but also the space of model parameters. Our goal is to eventually arrive at better selection and configuration of Machine Learning models using interactive visualization. We started our discussion with the question 'What is a model?' and developed a draft reference model for Model Space Visualization. To this end, we built on existing process models, including the Information Visualization model of Card, MacKinley and Shneiderman and the Visual Analytics model proposed by Thomas and Keim. Our model adds one level of detail to the formalism and distinguishes between expert and user roles. In particular, this new process model makes the integration of Machine Learning and Visualization explicit. Consideration of model instances and parameter sets as part of the workflow in our model aims at a tighter integration of machine learning into the interactive analysis process. Also, the model is aiming as a reference structure to survey and classify existing works in Visual Analytics and Visual Data Mining. These latter points are seen as interesting future work.

4.3 Results of Working Group: Embedding techniques at the crossing of Machine Learning and Information Visualization

Michael Aupetit and John Lee

License  Creative Commons BY-NC-ND 3.0 Unported license
© Michael Aupetit and John Lee

4.3.1 Attendees

- Information Visualisation: D. Keim; L. Zhang
- Machine Learning: M. Verleysen; J.A.Lee; M. Aupetit; S. Kaski; J. Peltonen; L. van der Maaten; F.-M. Schleif

4.3.2 Emerging topics of interest

- from the Information Visualisation perspective

Getting trust from the analyst is fundamental to make embedding common visual analytics tools. For this, the projection must not change drastically if no strong changes occur in the data distribution, so questions are:

 - How to make embeddings robust to noise and outliers?
 - How to make embeddings stable adding new data points and against local optima and different initialisation ?

Interactivity is also a very important point in visual analytics,

 - How to deal with massive datasets in terms of speed and quantity of data to visualize?
 - How to link embeddings of different local subspaces to get better understanding of the data?

Understandability is another main issue with non linear embeddings (axes have no sense):

 - How to connect embeddings to the meaning of the original data features?
- from the Machine Learning perspective

Assessing Visual Analytic tools needs well defined tasks:

 - Can we define a taxonomy of tasks and data types that could benefit from embeddings?

Raw data have to be preprocessed before embedding:

 - Which kind of preprocessing has to be done before embedding?
 - Which kind of similarity measures make embeddings more efficient for which kind of task?
 - How to deal with discrete, non-Cartesian, missing data?


4.3.3 Intended actions

- Sharing of various data sets (InfoVis) and embedding methods (ML)
- Build a joint InfoVis/ML taxonomy
- Organizing a workshop and a tutorial on embeddings at the next IEEE VisWeek 2012 conference from which we can edit a special journal issue on this topic

We thank all the contributors to this group, and all the colleagues from the Dagstuhl seminar for fruitful discussions.

4.4 Results of Working Group: Evaluation


Ian Nabney, Dianne Cook, Brian D. Fisher, Andrej Gisbrecht, Hans Hagen, and Heike Hoffman

License  Creative Commons BY-NC-ND 3.0 Unported license
 © Ian Nabney, Dianne Cook, Brian D. Fisher, Andrej Gisbrecht, Hans Hagen, and Heike Hoffman
URL <http://db.tt/G0Ajn0V6>

Our outcomes were captured by Ian Nabney in a mindmap which can be found at the URL below as a .mm file (<http://db.tt/G0Ajn0V6>). These files can be opened in a number of applications including Freemind <http://freemind.sourceforge.net/>

4.5 Results of Working Group: Fast Machine Learning

Jörn Kohlhammer

License  Creative Commons BY-NC-ND 3.0 Unported license
 © Jörn Kohlhammer

This session discussed the topic of “Fast ML for interactive visualization” and what the different perspectives are in ML and Infovis/VA/visualization.

It turned out that the ML community in its various sub-communities is not focused per se on performance issues of their algorithms, at least not to the extent of trying to achieve real-time capabilities. The InfoVis and VA community on the other hand is actively looking for high-performance, automated methods that can be coupled with visualization techniques to include more and more data in an interactive analysis. Response times are very important for interactive techniques and such response times do not play a major role in many ML approaches.

There were several thoughts about the user influence on ML methods, which might be beneficial to the ML community. One can distinguish between an internal coupling of methods, where the user interactively influences the automated methods during run-time, or an external coupling, which focuses on the flexible ensemble of ML methods and visualization methods along a structured analysis workflow.

The outcome of the discussion was that it would be highly interesting for the visualization community to learn about the current extent of research in this direction, i.e. a more performance-driven view on current research in ML:

- What type of ML methods do exist?
- Which sub-communities (or which research groups specifically) work on high-performance ML approaches?
- What are these approaches in detail? What are their characteristics, scalability constraints, data types, etc.?
- How could we jointly work on coupling such approaches with InfoVis and VA? Are their existing joint efforts, best practices, examples?
- What are the plans and future work in these Vis-relevant areas?

Next steps:


Our idea was to plan a tutorial for VisWeek 2012 with the tentative title “A performance perspective on machine learning for visualization”, to be submitted by 30 April 2012. The tutorial could be a half day or full day tutorial, depending on the outcome of the next planning steps. There could be 4-5 speakers or even more, again depending on the structure.

The tutorial should give an overview of ML methods and go into detail on the high-performance methods (along the lines of the above questions), building a possible repertoire of ML methods for visualization.

The joint understanding is that the talks in the tutorial should be held by ML experts, but with strong involvement of visualization experts in the planning phase to make sure that the talks are targeted at and are adequate/educational for the visualization community.

4.6 Results of Working Group: Future analysis environments

Jean-Daniel Fekete

License  Creative Commons BY-NC-ND 3.0 Unported license
© Jean-Daniel Fekete

The current situation of data analysis environments can be summarized simply by saying that there are numerous environments each of which has very satisfied users that do not want to switch to another solution. The only reasonable solution to avoid duplications of efforts is therefore to have some form of interoperability between environments. This can be provided at:

- a library level, with difficulties induced by differences between programming languages;
- an export/import level using e.g. xml formats with difficulties related to encoding and similar issues;
- a component level via rpc or web services mechanisms.

While interoperability might save the day, it has its share of problems:


- speed and latency;
- data duplication;
- limitation of some of the environments.

While one environment could rule them all, this seems unlikely, and improving interoperability seems a simpler goal. This needs not only the ability to share data, but also the support of notifications (of changes) and of metadata. One possible plan would be:

- specify and implement a sharing mechanism for R, Matlab, Excel...
 - how to connect/disconnect to a shared datatable
 - how to load content lazily
 - how to emit and receive notifications
 - how to manage content consistency
 - etc.
- test it

4.7 Results of Working Group: Structured/relational data

Nathalie Villa-Vialaneix

License  Creative Commons BY-NC-ND 3.0 Unported license
© Nathalie Villa-Vialaneix

Structured and relational data have been discussed and several issues have been extracted:

- clustering issues: evaluating the quality/relevance of a clustering/cluster
- taking into account heterogeneous data: heterogeneous data could lead to different clusterings: put the user in the loop to help find a consensual user-driven clustering

- metric came from mathematics; use the users' suggestions to try to find a consensus among the human experts and use ML to extract a relevant metric that fits the users' suggestion
- how to find a relevant labelling for a cluster: give the user hints automatic help for labelling

Participants

- Daniel Archambault
University College Dublin, IE
- Michael Aupetit
Commissariat à l’Energie
Atomique – Gif-sur-Yvette, FR
- Michael Biehl
University of Groningen, NL
- Kerstin Bunte
Universität Bielefeld, DE
- Etienne Come
IFSTTAR – Noisy le Grand, FR
- Dianne Cook
Iowa State University, US
- Jean-Daniel Fekete
Université Paris Sud – Orsay, FR
- Brian D. Fisher
Simon Fraser Univ. – Surrey, CA
- Ksenia Genova
TU Dresden, DE
- Andrej Gisbrecht
Universität Bielefeld, DE
- Hans Hagen
TU Kaiserslautern, DE
- Barbara Hammer
Universität Bielefeld, DE
- Helwig Hauser
University of Bergen, NO
- Heike Hofmann
Iowa State University, US
- Ata Kaban
University of Birmingham, GB
- Samuel Kaski
Aalto University and University
of Helsinki
- Johannes Kehrner
VRVis – Wien, AT
- Daniel A. Keim
Universität Konstanz, DE
- Andreas Kerren
Linnaeus University – Växjö, SE
- Jörn Kohlhammer
Fraunhofer Institut –
Darmstadt, DE
- Bongshin Lee
Microsoft Res. – Redmond, US
- John A. Lee
Université Catholique de
Louvain, BE
- Marcus A. Magnor
TU Braunschweig, DE
- Florian Mansmann
Universität Konstanz, DE
- Ian Nabney
Aston Univ. – Birmingham, GB
- Jaakko Peltonen
Aalto University, FI
- Gabriele Peters
FernUniversität in Hagen, DE
- Nathalie Riche-Henry
Microsoft Res. – Redmond, US
- Fabrice Rossi
Université Paris I, FR
- Frank-Michael Schleif
Universität Bielefeld, DE
- Tobias Schreck
Universität Konstanz, DE
- Marc Strickert
Universität Marburg, DE
- Holger Theisel
Universität Magdeburg, DE
- Peter Tino
University of Birmingham, GB
- Laurens van der Maaten
TU Delft, NL
- Jarke J. Van Wijk
TU Eindhoven, NL
- Michel Verleysen
Université Catholique de
Louvain, BE
- Nathalie Villa-Vialaneix
Université Paris I, FR
- Thomas Villmann
Hochschule Mittweida, DE
- Daniel Weiskopf
Universität Stuttgart, DE
- Hadley Wickham
Rice University – Houston, US
- Leishi Zhang
Universität Konstanz, DE



Principles of Provenance

Edited by

James Cheney¹, Anthony Finkelstein², Bertram Ludäscher³, and Stijn Vansummeren⁴

1 University of Edinburgh, GB, jcheney@inf.ed.ac.uk

2 University College London, GB

3 University of California, Davis, US, ludaesch@ucdavis.edu

4 Université Libre de Bruxelles, BE, stijn.vansummeren@ulb.ac.be

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 12091 “Principles of Provenance”. The term “provenance” refers to information about the origin, context, derivation, ownership or history of some artifact. In both art and science, provenance information is crucial for establishing the value of a real-world artifact, guaranteeing for example that the artifact is an original work produced by an important artist, or that a stated scientific conclusion is reproducible.

Since it is much easier to copy or alter digital information than it is to copy or alter real-world artifacts, the need for tracking and management of provenance information to testify the value and correctness of digital information has been firmly established in the last few years.

As a result, provenance tracking and management has been studied in many settings, ranging from databases, scientific workflows, business process modeling, and security to social networking and the Semantic Web, but with relatively few interaction between these areas.

This Dagstuhl seminar has focused on bringing together researchers from the above and other areas to identify the commonalities and differences of dealing with provenance; improve the mutual understanding of these communities; and identify main areas for further foundational provenance research.

Seminar 26 February – 2 March, 2012 – www.dagstuhl.de/12091

1998 ACM Subject Classification D.2 Software Engineering, D.3 Programming Languages, H.1 Models and Principles, H.2 Database Management

Keywords and phrases Provenance, Lineage, Metadata, Trust, Repeatability, Accountability


Digital Object Identifier 10.4230/DagRep.2.2.84

1 Executive Summary

James Cheney

Bertram Ludäscher

Stijn Vansummeren

License  Creative Commons BY-NC-ND 3.0 Unported license
© James Cheney, Bertram Ludäscher, and Stijn Vansummeren

The term “provenance” refers to information about the origin, context, derivation, ownership or history of some artifact. In both art and science, provenance information is crucial for establishing the value of a real-world artifact, guaranteeing for example that the artifact is an original work produced by an important artist, or that a stated scientific conclusion is reproducible. Even in everyday situations, we unconsciously use provenance to judge the quality of an artifact or process. For example, we often decide what food to buy based on



Except where otherwise noted, content of this report is licensed under a Creative Commons BY-NC-ND 3.0 Unported license

Principles of Provenance, *Dagstuhl Reports*, Vol. 2, Issue 2, pp. 84–113

Editors: James Cheney, Anthony Finkelstein, Bertram Ludäscher, and Stijn Vansummeren



Dagstuhl Reports

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

freshness, origin and “organic” labels; and we decide whether or not to believe an online news article based on its source, author, and timeliness.

Maintaining good records of provenance that are sufficient to convince skeptics of the value of an artifact is difficult. It requires reflection or monitoring actions as they are performed. Every step in the chain of ownership of an important work of art needs to be recorded in a secure way, for example, in order to defend against forgery and deter attempts to sell stolen artwork.

Since it is much easier to copy or alter digital information than to alter real-world artifacts, there are even more opportunities for misinformation, forgery and error in the digital world than there are in the traditional physical world. For this reason, the need for provenance is now widely appreciated. Simple and unreliable forms of automatic provenance tracking, such as version numbering, ownership, creation and modification timestamps in file systems, have long been supported as a basic services on which more sophisticated tools can rely. In today’s increasingly networked and decentralized world, however, we anticipate the need for richer provenance recording and management capabilities to be built into a wide variety of systems.

For example, “grid” or “cloud” computing infrastructures are frequently used for scientific computing, as part of a widespread trend towards “eScience”, “cyberinfrastructure” or more recently the data-intensive “fourth paradigm” of science popularized by Jim Gray and others. These systems are complex and opaque. The correctness and repeatability of scientific conclusions (about, for example, climate change) is increasingly being questioned because of the lack of transparency of the complex computer systems used to derive the results. Provenance technology can help to restore transparency and increase the robustness of eScience, countering increasing skepticism of scientific results as evidenced by the so-called “Climategate” controversy in 2009.

This problem is already widely appreciated in scientific settings but is increasingly recognized as a problem in business, industrial and Web settings. Until recently, work on provenance has mostly taken place in relatively isolated parts of existing research communities, such as databases, scientific workflow-based distributed computing, or file systems, or the Semantic Web. However, we believe that to make real progress it will be necessary to form a broader research community focusing on provenance.

In this respect, the aims of Dagstuhl Seminar 12091 “Principles of Provenance” were to:

- bring together researchers from databases, security, scientific workflows, software engineering, programming languages, and other areas to identify the commonalities and differences of provenance in these areas;
- improve the mutual understanding of these communities;
- identify main areas for further foundational provenance research.

The seminar hosted 41 participants in total from the above communities, and included representatives from the W3C Provenance Working group that is in the process of standardizing a common data model for representing and exchanging provenance information.

To improve the mutual understanding of the various communities, the first day of the seminar was devoted to tutorial talks from well-respected members of each community. An overview of these tutorials may be found in the Section “Overview of Tutorials” starting on p. 88.

The rest of the seminar consisted of presentations of recent ongoing provenance research in the various communities, as well as break-out sessions aimed at deepening discussions and identifying open problems. An overview of the talks may be found starting on p. 93. An overview of the breakout sessions may be found starting on p. 102. A list of open problems may be found on p. 105.

2 Table of Contents

Executive Summary

<i>James Cheney, Bertram Ludäscher, and Stijn Vansummeren</i>	84
---	----

Overview of Tutorials

Tutorial: Provenance in Databases <i>Wang-Chiew Tan, Todd J. Green, Chris Ré</i>	88
Tutorial: Provenance in Scientific Workflows <i>Bertram Ludäscher, Shawn Bowers, Paolo Missier</i>	89
Tutorial: Software Engineering, Programming Languages and Security Perspectives <i>Perdita Stevens, Steve Chong, James Cheney</i>	91
Highlights of W3C Provenance Incubator Group and Subsequent WG Activities <i>Luc Moreau, Paul Groth, Simon Miles</i>	92

Overview of Talks

Computation Slices as (Universal) Provenance <i>Umut A. Acar</i>	93
Engineering Options for Better Provenance Capture <i>Adriane Chapman</i>	93
Semantics of the PROV data model <i>James Cheney</i>	93
The Multi-granularity, Multi-Provenance (MMP) Model for Relational Databases <i>Lois Delcambre</i>	94
Using Provenance to enable Reproducible Science <i>Juliana Freire</i>	94
A new Approach for Publishing Workflows: Abstractions, Standards and Linked Data <i>Daniel Garijo</i>	95
The PROV-O Ontology <i>Daniel Garijo</i>	96
On the semantics of SPARQL on annotated RDF <i>Floris Geerts</i>	96
An Overview on W3C PROV-AQ: Provenance Access and Query <i>Olaf Hartig</i>	96
Modelling provenance using Structured Occurrence Networks <i>Paolo Missier</i>	97
The W3C PROV Provenance Data Model <i>Luc Moreau</i>	98
Tracing Where and Who Provenance in Linked Data: A Calculus <i>Vladimiro Sassone</i>	98
Toward Provenance as Cross-cutting Concern <i>Martin Schäler</i>	99

Self-Identifying Sensor Data	
<i>Christian Skalka</i>	100
When-provenance: Tracing the history and evolution of data	
<i>Wang-Chiew Tan</i>	100
Temporal semantics for the open provenance model	
<i>Jan Van den Bussche</i>	101
Cracking the quality jigsaw puzzle using provenance pieces – A speculation, not a solution	
<i>Jun Zhao</i>	101
Working Groups	
Formal models for provenance	
<i>Jan Van den Bussche</i>	102
Systems and security perspectives on provenance	
<i>Nate Foster</i>	103
Social Aspects of Provenance	
<i>Adriane Chapman</i>	103
Additional discussions	104
Open Problems	
Problems related to formal provenance models	105
Provenance, security, and confidentiality	108
Social Aspects of Provenance	111
Participants	113


3 Overview of Tutorials

3.1 Tutorial: Provenance in Databases

Wang-Chiew Tan (IBM Research & University of California, Santa Cruz, US)

Todd J. Green (University of California, Davis, US)

Chris Ré (University of Wisconsin-Madison, US)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Wang-Chiew Tan, Todd J. Green, Chris Ré

Various kinds of provenance have been defined in the database research community to give a very fine-grained account of the derivation of a piece of data appearing in the output of a database transformation, often a database query [1].

In general we can discern two kinds of approaches: *annotation-based approaches* and *non-annotation-based approaches*. Annotation-based approaches, also called eager approaches, explicitly record information about the derivation of a piece of data in the database itself, typically as an extra attribute in the table. Annotation-based approaches hence require that an annotation representing the provenance of a data item be recorded directly in the database and further require that the annotation be correctly propagated through future database transformations.

Non-annotation-based approaches, also called lazy approaches, in contrast, do not store provenance in the database, but analyze the query answer, the query itself, and the input tables to calculate the provenance of a piece of data. An example of non-annotation-based approach is *why-provenance* (which indicates the source tables that contributed a distinguished output tuple). An example of annotation-based approaches is *where-provenance* (which indicates where in the source database the piece of data was copied from).

How-provenance is an annotation-based approach that goes beyond why-provenance and where-provenance to capture the way in which data items (i.e., tuples) are combined to produce output items (i.e., query result). How-provenance annotations are typically represented using *provenance polynomial expressions* drawn from a semiring, as defined by the work of Green et al. [2]. The tutorial discussed all of these forms of provenance in detail, and illustrated in particular how the provenance polynomial approach to recording how-provenance plays a crucial role in a practical system: the Hazy statistical data processing system [3].

References


- 1 J. Cheney, L. Chiticariu, W. C. Tan Provenance in Databases: Why, How, and Where. *Foundations and Trends in Databases* 1(4), p. 379–474.
- 2 T. J. Green, G. Karvounarakis, V. Tannen. Provenance semirings. *PODS 2007*, p. 31–40.
- 3 C. Ré et al. Hazy: Analyzing Data from More Sources, More Deeply than Ever Before. <http://research.cs.wisc.edu/hazy/>

3.2 Tutorial: Provenance in Scientific Workflows

Bertram Ludäscher (University of California, Davis, US)

Shawn Bowers (Gonzaga University, US)

Paolo Missier (Newcastle University, GB)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Bertram Ludäscher, Shawn Bowers, Paolo Missier

As the natural sciences have become increasingly computational and data-driven,¹ scientific workflows have become popular as a means for scientists to automate computational pipelines, to take advantage of parallel platforms (clusters and clouds), and—last not least—to keep track of data lineage and other provenance information to facilitate *reproducible science*. The tutorial was structured into three parts: (i) Overview: Introduction to scientific workflows and provenance (presented by Bertram Ludäscher); (ii) Technical challenges for managing provenance data from scientific workflows (presented by Shawn Bowers); and (iii) Overview of current research strands in workflow-based provenance (presented by Paolo Missier).

A *scientific workflow* is the description of a process for accomplishing a scientific objective, usually expressed in terms of tasks and their dependencies. Typically, scientific workflow tasks are computational steps for scientific simulations or data analysis steps. Common elements or stages in scientific workflows are acquisition, integration, reduction, visualization, and publication (e.g., in a shared database) of scientific data [5]. Scientific workflows share commonalities with business workflows and business process management approaches, but there are significant differences as well: e.g., the former are data-centric and often use a dataflow execution model, while the latter focus on processes and control-flow; scientific workflows emphasize scalable, automated execution [4], while workflow modeling and analysis are often the focus in business process management [6]. Scientific workflow systems (such as Askalon, Kepler, Pegasus, Taverna, VisTrails, etc.) provide a controlled execution environment for executing computational pipelines and thus offer unique opportunities to capture provenance information [2, 3], which can be used subsequently to explain or “debug” workflow results.

The opportunities to capture detailed provenance information in scientific workflows give rise to a number of technical challenges associated with storing, querying, and presenting (visualizing) scientific workflow provenance information (e.g., see [1, 8]). Some of these issues were presented in the second part of the tutorial, using examples and solutions from the Kepler workflow system.

Standards such as the Open Provenance Model (OPM) [10], which resulted from a community effort starting with the First Provenance Challenge workshop [11], are designed to provide a least common denominator, and thus by design do not include aspects specific to scientific workflow provenance.² As a result, provenance interoperability (e.g., see [7]) remains an important research topic, in particular, when taking into account fine-grained and “precise” provenance in the presence of different execution models, data models, and provenance models of the underlying workflow systems.

In the last part of the tutorial, a high-level taxonomy of research strands in the area of provenance for workflow-based applications was presented [9]. Its main branches are (i)

¹ This is witnessed, e.g., by notions such “e-Science”, the “4th Paradigm” (i.e., data-driven scientific discovery, with the 3rd Paradigm being “simulation/computational science”), and “Big Data”.

² A scientific workflow-centric extension of OPM is under development by the DataONE (dataone.org) Working Group on Provenance in Scientific Workflows.

modelling, (ii) *capturing*, (iii) *exploiting* provenance. Each branch contains a number of bibliographic references (occasionally commented) as its leaves.

The “modelling” branch addresses the topic of the convergence between database and process-based provenance, as well as the emerging research on privacy-preserving provenance, and “human in the loop” provenance. Each of these topics were perceived as increasingly important by the seminar participants. Amongst the main issues in the “capturing” branch are (i) provenance for non-workflow processes, mainly scripting languages for science; (ii) virtual experiments, represented by multiple semi-independent provenance traces; (iii) system-level provenance, and (iii) how to make provenance secure, tamper-evident, and trustworthy. Finally, the “exploitation” branch includes (i) provenance analytics, (ii) Provenance for reproducibility, and (iii) Provenance for improving data engineering. The index [9] is meant to be periodically updated, to form a more comprehensive reference for researchers in this particular area of provenance studies.

References


- 1 M.K. Anand, S. Bowers, and B. Ludäscher. Techniques for efficiently querying scientific workflow provenance graphs. In *Intl. Conf. on Extending Database Technology (EDBT)*, pages 287–298. ACM, 2010.
- 2 S. Davidson, S.C. Boulakia, A. Eyal, B. Ludäscher, T.M. McPhillips, S. Bowers, M.K. Anand, and J. Freire. Provenance in scientific workflow systems. *IEEE Data Eng. Bull.*, 30(4):44–50, 2007.
- 3 S.B. Davidson and J. Freire. Provenance and scientific workflows: challenges and opportunities. In *SIGMOD conference*, pages 1345–1350. ACM, 2008.
- 4 Y. Gil, E. Deelman, M. Ellisman, T. Fahringer, G. Fox, D. Gannon, C. Goble, M. Livny, L. Moreau, and J. Myers. Examining the challenges of scientific workflows. *Computer*, 40(12):24–32, 2007.
- 5 B. Ludäscher, S. Bowers, and T. McPhillips. Scientific workflows. In Ling Liu and M. Tamer Özsu, editors, *Encyclopedia of Database Systems*, pages 2507–2511. Springer, 2009.
- 6 B. Ludäscher, M. Weske, T. McPhillips, and S. Bowers. Scientific workflows: Business as usual? In *Intl. Conf. on Business Process Management (BPM)*, pp. 31–47, 2009. LNCS 5701.
- 7 P. Missier, B. Ludäscher, S. Bowers, S. Dey, A. Sarkar, B. Shrestha, I. Altintas, M.K. Anand, and C. Goble. Linking multiple workflow provenance traces for interoperable collaborative science. In *5th Workshop on Workflows in Support of Large-Scale Science (WORKS)*, New Orleans, 2010.
- 8 P. Missier, N.W. Paton, and K. Belhajjame. Fine-grained and efficient lineage querying of collection-based workflow provenance. In *Intl. Conf. on Extending Database Technology (EDBT)*, pages 299–310. ACM, 2010.
- 9 Paolo Missier. Research strands in workflow-based provenance. homepages.cs.ncl.ac.uk/paolo.missier/doc/Dagstuhl-PoP/Research_strands.html, 2012.
- 10 L. Moreau, B. Clifford, J. Freire, J. Futrelle, Y. Gil, P. Groth, N. Kwasnikowska, S. Miles, P. Missier, J. Myers, et al. The open provenance model core specification (v1. 1). *Future Generation Computer Systems*, 27(6):743–756, 2011.
- 11 L. Moreau, B. Ludäscher, I. Altintas, R.S. Barga, S. Bowers, S. Callahan, G. Chin Jr, B. Clifford, S. Cohen, S. Cohen-Boulakia, et al. Special issue: The first provenance challenge. *Concurrency and Computation: Practice and Experience*, 20(5):409–418, 2008.

3.3 Tutorial: Software Engineering, Programming Languages and Security Perspectives

Perdita Stevens (University of Edinburgh, GB)

Steve Chong (Harvard University, US)

James Cheney (University of Edinburgh, GB)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Perdita Stevens, Steve Chong, James Cheney

This tutorial touched upon three distinct themes: provenance in software engineering (presented by Perdita Stevens); provenance in programming languages (presented by James Cheney); and provenance and security (presented by Steve Chong).

From the earliest days of software engineering, practitioners have been concerned to trace the connections between the requirements that a software system must satisfy and the tests that establish that requirements have been met. This is termed *traceability*, and the same term is then used much more broadly in software engineering could be called provenance. Traceability is typically recorded as a so-called *requirements traceability matrix*, which is formally a binary relation on Requirements and Tests. Even with the best available commercial tool support, maintaining traceability information is a time-consuming partly manual process. It has been repeatedly observed that in practice, this maintenance is not well done. This is not (always) laziness on the part of the developers: the cost/benefit ratio often does not favour doing so. Moreover, the traceability information that is maintained may not be the information that is most needed. It has been reported that most traceability problems require tracing back before the development of the requirements specification which is typically the beginning of the traceability process. If provenance information is to be more widely collected and used it will be important to avoid reproducing these problems. Specifically, it is notable that the above gives, as yet, no common definition of what provenance information, annotation or traces mean, outside the pleasant world of databases.

In programming languages research, a number of sophisticated techniques have been proposed to track and control the flow of information in systems. In this tutorial, these techniques were motivated and explained. Subsequently, it was shown how information flow control techniques could be used to enforce security, and how this links with provenance.

Other concepts related to provenance in programming languages range from simple conveniences such as source code line number information used in compilers, to program slicing (a classical debugging technique widely studied in imperative programming languages), algorithmic debugging, type inference and type error slicing, dependency tracking, and language-based security. This tutorial covered some recent developments in formalizing security properties for provenance, including the properties of disclosure and obfuscation [1]. Additional topics, such as self-adjusting computation, bidirectional programming, and blame and contracts also seem relevant but to date there has been little work relating them and provenance.

References


- 1 J. Cheney. A formal framework for provenance security. In *CSF*, pages 281–293. IEEE, 2011.
- 2 J. Cheney, A. Ahmed, and U. A. Acar. Provenance as dependency analysis. *Mathematical Structures in Computer Science*, 21(6):1301–1337, 2011.

3.4 Highlights of W3C Provenance Incubator Group and Subsequent WG Activities

Luc Moreau (University of Southampton, GB)

Paul Groth (VU University Amsterdam, NL)

Simon Miles (King's College London, GB)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Luc Moreau, Paul Groth, Simon Miles

In 2009, a W3C Provenance Incubator Group³ was charged with the task of providing a state-of-the art understanding of provenance for Semantic Web technologies, and developing a roadmap for development, and possible standardization of such technologies.

Based on the conclusions of the Incubator group, the W3C Provenance Working Group⁴ is currently in the process of defining a family of standards for the representation, exchange, location, and querying of provenance information on the web.

This tutorial gives an overview of the conclusions of the W3C Provenance Incubator group, as well as an overview of the standards that are currently under definition:

- PROV-DM [1] is a data model for provenance that describes the entities, people and activities involved in producing a piece of data or thing in the world. PROV-DM is domain-agnostic, but is equipped with extensibility points allowing further domain-specific and application-specific extensions to be defined.
- PROV-DM is accompanied by PROV-N [2], a technology-independent notation, which allows serializations of PROV-DM instances to be created for human consumption, which facilitates the mapping of PROV-DM to concrete syntax, and which is used as the basis for a formal semantics of PROV-DM that is currently under development.
- PROV-DM is also accompanied by PROV-O [3], a translation of PROV-DM into an OWL ontology for the purpose of expression of provenance in RDF.
- Finally, PROV-AQ [4] specifies how one can use standard Web protocols, including HTTP, to obtain information about the provenance of Web resources. It describes both simple access mechanisms for locating provenance information associated with web pages or resources, as well as provenance query services for more complex deployments.

References

- 1 The Provenance Data Model. L. Moreau, P. Missier (eds.) K. Belhajjame, S. Cresswell, Y. Gil, R. B'Far, P. Groth, G. Klyne, J. McCusker, S. Miles, J. Myers, S. Sahoo. W3C Working Draft, 2012. <http://www.w3.org/TR/prov-dm/>
- 2 The PROV Data Model and Abstract Syntax Notation. L. Moreau, P. Missier (eds.), K. Belhajjame, S. Cresswell, Y. Gil, R. Golden, P. Groth, G. Klyne, J. McCusker, S. Miles, J. Myers, S. Sahoo. W3C Working Draft, 2012. <http://www.w3.org/TR/prov-n/>
- 3 The PROV Ontology: Model and Formal Semantics. S. Sahoo, D. McGuinness (eds.) K. Belhajjame, J. Cheney, D. Garijo, T. Lebo, S. Soiland-Reyes, S. Zednik. W3C Working Draft, 2012. <http://www.w3.org/TR/prov-aq/>
- 4 Provenance Access And Query. L. Moreau, P. Groth (eds.), O. Hartig, Y. Simmhan, J. Myers, T. Lebo, K. Belhajjame, S. Miles. W3C Working Draft, 2012. <http://www.w3.org/TR/prov-o/>


³ http://www.w3.org/2005/Incubator/prov/wiki/W3C_Provenance_Incubator_Group_Wiki

⁴ http://www.w3.org/2011/prov/wiki/Main_Page

4 Overview of Talks

4.1 Computation Slices as (Universal) Provenance

Umut A. Acar (MPI for Software Systems – Kaiserslautern, DE)


License  Creative Commons BY-NC-ND 3.0 Unported license
© Umut A. Acar

Joint work of Acar, Umut A.; Cheney, James; Levy Paul; Perera, Roly

I present techniques that enable higher-order functional computations to “explain” their work by answering questions about how parts of their output were calculated. As explanations, I consider the traditional notion of program slices, which can be inadequate, and propose a new notion: computation slices. I present techniques for specifying flexible and rich slicing criteria based on partial expressions part of which are replaced by holes and present an “unevaluation” algorithm, for computing least program slices from computations reified as traces. In addition, I define the notion of a computation slices and briefly describe how they minimal computation slices can be computed.

4.2 Engineering Options for Better Provenance Capture

Adriane Chapman (MITRE – McLean, US)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Adriane Chapman

Joint work of Chapman, Adriane; Allen, David; Blaustein, Barbara; Seligman, Len


Main reference M. D. Allen, A. Chapman, B. Blaustein, L. Seligman, “Provenance Capture in the Wild,” in Proc. of Third Int’l Provenance and Annotation Workshop (IPAW’10), pp. 98–101, LNCS, vol. 6378, 2010.

URL http://dx.doi.org/10.1007/978-3-642-17819-1_12

The research literature contains a fair amount of work about the positive things that can be done with provenance information. All of them though start with the presumption that a system actually has provenance information, which simply is not the case for most systems today. The value of provenance cannot be realized without first capturing it. While most of the literature further assumes central control over the a monolithic system in question (for example, a biomed researcher capturing provenance about their own experimental setup) most systems in the wild are neither centrally controlled nor monolithic in their technology selection. This talk addresses the many options and strategies for capturing provenance in real, large IT systems along with their pros and cons.

4.3 Semantics of the PROV data model

James Cheney (University of Edinburgh)


License  Creative Commons BY-NC-ND 3.0 Unported license
© James Cheney

The W3C PROV data model is based on an intuition that provenance information records a history of entities, activities, agents and interactions among them. A central and subtle issue is the fact that entities change over time, and the properties we use to describe them may not be fixed. To untangle these issues, the W3C group has been developing a formal

semantics, that is a mathematical model, with respect to which we can assign meanings to PROV statements, thinking of instances of the PROV data model as collections of logical statements describing some past events. In particular, PROV includes relations between different versions of the same entity at different times, or between more and less specific aspects of the same entity. The talk presented the semantics, focusing on these special relations and the underlying mathematical framework that helps explain their properties.

4.4 The Multi-granularity, Multi-Provenance (MMP) Model for Relational Databases

Lois Delcambre (Portland State University)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Lois Delcambre

Joint work of Archer, David; Lois Delcambre

Main reference D. Archer, “Conceptual Modeling of Data with Provenance,” PhD Dissertation, Computer Science Department, Portland State University, 2011, (Lois Delcambre, advisor)

URL <http://www.pdx.edu/sites/www.pdx.edu/computer-science/files/archerthesis2011.pdf>

In a relational database setting, the main interactions with the database are using the insert, update, delete, and query operators. Historically, databases systems with provenance have considered mechanisms to track tuples that produce query answers, e.g., as described by polynomials of various kinds (e.g., based on the work of Todd Green). In this talk, we’ll present a conceptual model for provenance in databases where the database system records all provenance explicitly, at a detailed level for all of the above operators. The database user can easily browse forward and backward through provenance and can issue queries to find current data based on characteristics of the provenance. Features of this model include that we track provenance for values, tuples, attributes, and tables (multi-granularity) and that we allow values in a database to have multiple provenances, e.g., from multiple insertions.

4.5 Using Provenance to enable Reproducible Science

Juliana Freire (Polytechnic Institute of NYU – Brooklyn, US)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Juliana Freire

Joint work of Freire, Juliana; David Koop; Emanuele Santos; Huy T. Vo; Philippe Bonnet; Matthias Troyer; Claudio Silva

URL <http://www.vistrails.org>

Important scientific results give insight and lead to practical progress. The ability to test these results is crucial for science to be self-correcting, and the ability to re-use and extend the results is key for science to move forward. In natural science, long tradition requires that results be reproducible, and in math, formal proofs that can be verified must accompany results. However, the same standard has not been applied for results backed by computational experiments.

Most computational experiments are specified only informally in papers, where experimental results are briefly described in figure captions; the code that produced the results is seldom available; and configuration parameters change results in unforeseen ways. The lack of reproducibility for computational results currently reported in the literature has raised questions about their reliability and has led to a widespread discussion on the importance of

computational reproducibility. However, a major barrier to a wider adoption of reproducibility is the fact that it is hard both for authors to derive a compendium that encapsulates all the components (e.g., data, code, parameter settings, environment) needed to reproduce a result, and for reviewers to verify the results.


As a step towards simplifying the creation and review of reproducible results, and motivated by the needs of computational scientists, we have built an infrastructure that supports the life cycle of computational experiments. This infrastructure makes it easier to generate and share repeatable results by making provenance a central component in scientific exploration, and the conduit for integrating data acquisition, derivation, and analysis as executable components throughout the publication process. Provenance is systematically and transparently captured and it includes all meta-data necessary to reproduce experiments, including the specifications of the computations, input and output data, source code, and library versions. We have also developed a set of solutions to address practical aspects related to reproducibility, including methods to link results to their provenance, explore parameter spaces, wrap command-line tools, interact with results through a Web-based interface, and upgrade the specification of computational experiments to work in different environments and with newer versions of software. This infrastructure has been implemented and released as part of VisTrails (<http://www.vistrails.org>), an open-source workflow-based data exploration and visualization tool, and it is already being used by different groups of scientists. Videos that illustrate the process to create reproducible publications using VisTrails are available at <http://www.vistrails.org/index.php/RepeatabilityCentral>.

References

- 1 D. Koop, E. Santos, P. Mates, H. T. Vo, P. Bonnet, B. Bauer, B. Surer, M. Troyer, D. N. Williams, J. E. Tohline, J. Freire, and C. T. Silva. A provenance-based infrastructure to support the life cycle of executable papers. *Procedia Computer Science*, 4:648–657, 2011. Proceedings of the International Conference on Computational Science, ICCS 2011.

4.6 A new Approach for Publishing Workflows: Abstractions, Standards and Linked Data

Daniel Garijo (Universidad Politécnica de Madrid, ES)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Daniel Garijo

Joint work of Garijo, Daniel; Gil, Yolanda

Main reference D. Garijo, Y. Gil, “A new approach for publishing workflows: abstractions, standards, and linked data,” in Proc. of 6th Workshop on Workflows in Support of Large-Scale Science (WORKS’11), pp. 47–56, ACM, New York, NY, 2011.


URL <http://dx.doi.org/10.1145/2110497.2110504>

In recent years, a variety of systems have been developed that export the workflows used to analyze data and make them part of published articles. We argue that the workflows that are published in current approaches are dependent on the specific codes used for execution, the specific workflow system used, and the specific workflow catalogs where they are published. We take a new approach that addresses these shortcomings and makes workflows more reusable through: 1) the use of abstract workflows to complement executable workflows to make them reusable when the execution environment is different, 2) the publication of both abstract and executable workflows using standards such as the Open Provenance Model that can be imported by other workflow systems, 3) the publication of workflows as Linked Data that results in open web accessible workflow repositories. As part of this work, we developed

the OPMW profile for OPM that allows us to publish abstract workflows and link them to the workflow execution provenance. We illustrate this approach using a complex workflow that we re-created from an influential publication that describes the generation of ‘drugomes’.

4.7 The PROV-O Ontology

Daniel Garijo (Universidad Politécnica de Madrid, ES)


License  Creative Commons BY-NC-ND 3.0 Unported license
© Daniel Garijo

Joint work of Garijo, Daniel; Lebo Timothy; Sahoo, Satya; McGuinness, Deborah; Lang, Mike; Belhajjame, Khalid; Cheney, James; Soiland-Reyes, Stian; Zednik, Stephan; Zhao, Jun

In this short talk, I introduce the PROV-O Ontology, an OWL-RL mapping of the PROV Data model. In the presentation I explain briefly the main classes, relationships and the RDF serialization of a complete example.

4.8 On the semantics of SPARQL on annotated RDF

Floris Geerts (University of Edinburgh, GB)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Floris Geerts


Joint work of Cristophides, Vassilis; Fundulaki, Irini; Geerts, Floris; Karvounarakis, Grigoris

We revisit the semantics of SPARQL on RDF in the presence of annotations. It is readily verified that for such a semantics to work correctly, one needs operations on annotations that correspond to the various operators supported by SPARQL, and furthermore, these annotation operations need to adhere to certain algebraic identities. It readily follows that when the positive fragment of SPARQL is considered, a semiring structure on the annotations is required. Semirings, however, do not suffice when dealing with the OPTIONAL construct in SPARQL.

Instead, we identify a new algebraic structure for SPARQL annotations, define a corresponding free object and show how it can be used to evaluate SPARQL on annotated RDF.

4.9 An Overview on W3C PROV-AQ: Provenance Access and Query

Olaf Hartig (Humboldt Universität zu Berlin, DE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Olaf Hartig

Joint work of Klyne, Graham; Groth, Paul; Moreau, Luc; Hartig, Olaf; Simmhan, Yogesh; Myers, James; Lebo, Timothy; Belhajjame, Khalid; Miles, Simon;

Main reference L. Moreau, O. Hartig, Y. Simmhan, J. Myers, T. Lebo, K. Belhajjame, S. Miles, “PROV-AQ: Provenance Access and Query,” W3C Working Draft, 10 January 2012, edited by Graham Klyne and Paul Groth.

URL <http://www.w3.org/TR/prov-aq/>

This short talk introduces the “Provenance Access and Query” (PAQ) document which is part of the PROV family of documents developed by the W3C Provenance Working Group.

The purpose of PAQ is to describe how to locate, retrieve, and query provenance information on the Web. The talk will briefly introduce the following main contributions of PAQ:

- A simple mechanism for discovery and retrieval of provenance information; and
- More advanced discovery service and query mechanisms.

Finally, we will point out some of the open issues of the current version of PAQ.

4.10 Modelling provenance using Structured Occurrence Networks

Paolo Missier (Newcastle University, GB)

License  Creative Commons BY-NC-ND 3.0 Unported license

© Paolo Missier

Joint work of Missier, Paolo; Randell, Brian; Koutny, Maciej

Main reference P. Missier, B. Randell, M. Koutny, “Modelling Provenance using Structured Occurrence Networks,” in Proc. of 4th Int’l Provenance and Annotation Workshop (IPAW’12), Santa Barbara, CA, June 2012.

URL <http://homepages.cs.ncl.ac.uk/paolo.missier/doc/Dagstuhl-SON-provenance.pdf>


Occurrence Nets (ON) are directed acyclic graphs that represent causality and concurrency information concerning a single execution of a system. Structured Occurrence Nets (SONs) extend ONs by adding new relationships, which provide a means of recording the activities of multiple interacting, and evolving, systems. Although the initial motivations for their development focused on the analysis of system failures, their structure makes them a natural candidate as a model for expressing the execution traces of interacting systems. These traces can then be exhibited as the provenance of the data produced by the systems under observation. In this paper we present a number of patterns that make use of SONs to provide principled modelling of provenance. We discuss some of the benefits of this modelling approach, and briefly compare it with others that have been proposed recently. SON-based modelling of provenance combines simplicity with expressiveness, leading to provenance graphs that capture multiple levels of abstraction in the description of a process execution, are easy to understand and can be analyzed using familiar graph query techniques.

References

- 1 E. Best and R. Devillers. Sequential and concurrent behaviour in Petri net theory. *Theoretical Computer Science*, 55(1):87–136, 1987.
- 2 D. Harel and P. Thiagarajan. Message Sequence Charts. In L. Lavagno, G. Martin, and B. Selic, editors, *UML for Real*, pages 77–105. Springer US, 2004.
- 3 J. Kleijn and M. Koutny. Causality in Structured Occurrence Nets. In C. Jones and J. Lloyd, editors, *Dependable and Historic Computing*, volume 6875 of *Lecture Notes in Computer Science*, pages 283–297. Springer Berlin / Heidelberg, 2011.
- 4 M. Koutny and B. Randell. Structured Occurrence Nets: A Formalism for Aiding System Failure Prevention and Analysis Techniques. *Fundamenta Informaticae*, 97, 2009.
- 5 B. Randell. Occurrence Nets Then and Now: The Path to Structured Occurrence Nets. In L. Kristensen and L. Petrucci, editors, *Applications and Theory of Petri Nets*, volume 6709 of *Lecture Notes in Computer Science*, pages 1–16. Springer Berlin / Heidelberg, 2011.

4.11 The W3C PROV Provenance Data Model

Luc Moreau (University of Southampton, GB)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Luc Moreau

Joint work of Moreau, Luc; Paolo Missier; Belhajjame, Khalid; Cresswell, Stephen; Gil, Yolanda; B'Far, Reza; Groth, Paul; Klyne, Graham; McCusker, Jim; Miles, Simon; Myers, James; Sahoo, Satya

Main reference L. Moreau, P. (eds.) – K. Belhajjame, R. B'Far, S. Cresswell, Y. Gil, P. Groth, G. Klyne, J. McCusker, S. Miles, J. Myers, S. Sahoo, (contributors), “The Provenance Data Model,” W3C Working Draft, 03 February 2012.

URL <http://www.w3.org/TR/prov-dm/>

PROV-DM is a data model for provenance that describes the entities, people and activities involved in producing a piece of data or thing in the world. PROV-DM is domain-agnostic, but is equipped with extensibility points allowing further domain-specific and application-specific extensions to be defined.


PROV-DM is accompanied by PROV-N, a technology-independent notation, which allows serializations of PROV-DM instances to be created for human consumption, which facilitates the mapping of PROV-DM to concrete syntax, and which is used as the basis for a formal semantics of PROV-DM.

References

- 1 The Provenance Data Model. L. Moreau, P. Missier (eds.) K. Belhajjame, S. Cresswell, Y. Gil, R. B'Far, P. Groth, G. Klyne, J. McCusker, S. Miles, J. Myers, S. Sahoo. W3C Working Draft, 02 February 2012. <http://www.w3.org/TR/2012/WD-prov-dm-20120202/>. Latest version: <http://www.w3.org/TR/prov-dm/>

4.12 Tracing Where and Who Provenance in Linked Data: A Calculus

Vladimiro Sassone (University of Southampton, GB)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Vladimiro Sassone

Joint work of Dezani-Ciancaglini, Mariangiola; Horne, Ross; Sassone, Vladimiro

Main reference M. Dezani, R. Horne, V. Sassone, “Tracing where and who provenance in Linked Data: a calculus,” Theoretical Computer Science, *in press*. Pre-print available.

URL <http://eprints.soton.ac.uk/335248/>

Linked Data provides some sensible guidelines for publishing and consuming data on the Web. Data published on the Web has no inherent truth, yet its quality can often be assessed based on its provenance.

This work introduces a new approach to provenance for Linked Data. The simplest notion of provenance – viz., a named graph indicating where the data is now – is extended with a richer provenance format. The format reflects the behaviour of processes interacting with Linked Data, tracing where the data has been published and who published it. An executable model is presented based on abstract syntax and operational semantics, providing a proof of concept and the means to statically evaluate provenance driven access control using a type system.

4.13 Toward Provenance as Cross-cutting Concern

Martin Schäler (Universität Magdeburg, DE)

License © ⓘ ⓘ Creative Commons BY-NC-ND 3.0 Unported license
© Martin Schäler

Joint work of Schäler, Martin; Schulze, Sandro; Saake, Gunter

Main reference M. Schäler, S. Schulze, G. Saake, “A Hierarchical Framework for Provenance Based on Fragmentation and Uncertainty,” Technical Report FIN-01-2012, School of Computer Science, University of Magdeburg, Germany, 2012.

URL http://www.witi.cs.uni-magdeburg.de/iti_db/publikationen/ps/auto/SSS2012.pdf

Provenance gained much attention in the recent past, especially for explaining and validating origin as well as derivation history of data. Furthermore, this term is used in many communities such as fine-grained annotations in relational databases, domain-specific approaches in scientific workflows, and even to determine source code ownership. Thus, we cannot give a clear definition about provenance sufficient for all these communities. In fact, it is even hard to give a clear definition to one of these communities. As a result, current solutions capturing provenance, are not sufficient in complex systems (e.g., forensics, medical data) where data items cross the borders of multiple systems, having different granularities, or even non computational steps are involved.

We argue that creating a solution for every application domain covering the versatile characteristics of provenance is inflexible, laborious, or even impossible. In contrast, our vision is to integrate provenance as cross-cutting concern into existing systems efficiently. As first step to realize our vision, we analyzed the literature addressing different parts of provenance to identify commonalities in provenance. Based on the current state of the art, there are three characteristics that seem to hold generally for provenance information [1]. Provenance is unchangeable, fragmentary at different levels of granularity, and contains a certain amount of uncertainty. While the first characteristic is a fundamental prerequisite the latter ones are dimensions of provenance, allowing to build a hierarchical framework covering a broad variety of approaches reaching from coarse grained notations (e.g., Open Provenance Model) to the principles of fine grained formal approaches (e.g., why and where provenance, semiring model). Furthermore, we use this framework to differentiate between provenance and related fields such as causality.

Currently, we started to integrate the cross-cutting provenance concern, based on our framework, into existing systems. Therefore, we analyze the feasibility of applying techniques from modern software engineering allowing a minimal invasive integration and if necessary un-integration of provenance. Furthermore, we evaluate their advantages and drawbacks. As a starting point we have chosen database systems, because there are formal models which can be implemented and recent insights such extensions of the semiring model for aggregate queries and linking provenance to causality seem to be promising to apply parts of the solutions to different data models and programming paradigms. Finally, linking different systems where we capture provenance (in a reliable way) is another important challenge. To this end, we propose the use of invertible watermarking schemes tailored to the requirements of the underlying systems [2].

For the future, we aim at identifying open research issues and present respective solutions, to move the borders hindering to fulfill our vision of provenance as cross-cutting concern.


References

- 1 M. Schäler, S. Schulze, and G. Saake. *A Hierarchical Framework for Provenance Based on Fragmentation and Uncertainty*. Technical Report FIN-01-2012, School of Computer Science, University of Magdeburg, Germany, 2012

- 2 M. Schäler, S. Schulze, R. Merkel, G. Saake, and J. Dittmann. *Reliable Provenance Information for Multimedia Data Using Invertible Fragile Watermarks*. 28th British National Conference on Databases (BNCOD), volume 7051 of LNCS, pages 3–17. Springer, 2011

4.14 Self-Identifying Sensor Data

Christian Skalka (University of Vermont, US)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Christian Skalka

Joint work of Skalka, Christian; Chong, Stephen; Vaughan, Jeffrey

Main reference S. Chong, C. Skalka, J.A. Vaughan, “Self-Identifying Sensor Data,” in Proc. of 9th Int’l Conf. on Information Processing in Sensor Networks (IPSN’10), pp. 82–93, ACM, 2010.

URL <http://dx.doi.org/10.1145/1791212.1791223>


Public-use sensor datasets are a useful scientific resource with the unfortunate feature that their provenance is easily disconnected from their content. To address this we introduce a technique to directly associate provenance information with sensor datasets. Our technique is similar to traditional watermarking but is intended for application to unstructured time-series datasets. Our approach is potentially imperceptible given sufficient margins of error in datasets, and is robust to a number of benign but likely transformations including truncation, rounding, bit-flipping, sampling, and reordering. We provide algorithms for both one-bit and blind mark checking, and show how our system can be adapted to various data representation types. Our algorithms are probabilistic in nature and are characterized by both combinatorial and empirical analyses.

References

- 1 S. Chong, C. Skalka, and J. Vaughan. *Self-Identifying Sensor Data*. In ACM Conference on Information Processing in Sensor Networks (IPSN), 2010.

4.15 When-provenance: Tracing the history and evolution of data

Wang-Chiew Tan (IBM Research & University of California – Santa Cruz, US)


License  Creative Commons BY-NC-ND 3.0 Unported license
© Wang-Chiew Tan

Many scientific, business, and Web datasets produced today are hierarchical and associated with multiple dimensions of time. Archiving such data in a way that preserves the semantics of the different time dimensions can help understand the past and anticipate the future. However, there have been very few systems that can effectively create a semantic archive of such evolving hierarchical data under more than one time dimension.

We have recently developed a system, called Tempura, that supports efficient and compact temporal archiving of evolving hierarchical data under multiple dimensions of time. Tempura creates a multi-dimensional longitudinal record of knowledge about an entity by grouping entities across different snapshots together in the archive. The associated time dimensions are coalesced and independently varied to maintain a consistent view of the entity over time. We call such multidimensional longitudinal knowledge of an entity its *when-provenance*, which intuitively corresponds to when one knows what one knows about the entity. I will describe how the Tempura archive model naturally captures when-provenance, its implementation, and how it can support temporal data visualization.

4.16 Temporal semantics for the open provenance model

Jan Van den Bussche (Hasselt University, BE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Jan Van den Bussche

Joint work of Kwasnikowska, Natalia; Moreau, Luc; Van den Bussche, Jan


Main reference N. Kwasnikowska, L. Moreau, J. Van den Bussche, “A Formal Account of the Open Provenance Model,” University of Southampton ECS Eprint 21819.

URL <http://eprints.ecs.soton.ac.uk/21819/>

The Open Provenance Model (OPM) is a graph-based data model for the representation of provenance information. Provenance information could be defined roughly as information about “what has happened” during some complex process. OPM is expected to heavily influence a W3C standard for provenance which is in the making. The current OPM specification defines a graph-based syntax, as well as some inference rules. A formal semantics that explains the soundness of these inference rules, and that could be used to prove completeness of the inference rules, was lacking however. In this paper we will propose a temporal semantics for OPM graphs; we will see that the current inference rules are, in fact, incomplete, and we will provide a complete set of inference rules.

4.17 Cracking the quality jigsaw puzzle using provenance pieces – A speculation, not a solution

Jun Zhao (University of Oxford, GB)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Jun Zhao


Joint work of Wf4Ever consortium

Digital science brings sea change to scientific research. A vast number of scientific data is made available in digital format, without their paper counterparts. Digital, or computational, experiments are increasingly used to replace or complement their wet-lab peers. ‘Big’ science becomes possible as scientists start to collaborate using data and methods shared and published on the Web. However, quality of data remains a major concern of scientists. The astronomy scientists we work with explicitly express their concern in trusting and reusing digital data, results and methods published and shared by third parties. To this end, we investigate the role of provenance information in producing a ‘quality stamp’ upon these research resources. We speculate different provenance pieces that can be drawn together. Instead of presenting solutions, we hope to stimulate further discussions regarding this topic.

5 Working Groups

5.1 Formal models for provenance

Led by James Cheney and Jan van den Bussche, and summarized by Jan Van den Bussche (Hasselt University, BE)

License  Creative Commons BY-NC-ND 3.0 Unported license
© Jan Van den Bussche

Joint work of All participants of the formal models for provenance break-out sessions.

There were two break-out sessions concerning formal models for provenance.

In the first session, we reviewed and commented on the current draft of the W3C PROV formal semantics. In particular, we reviewed the notion of *world model*, the notion of *object*, and the connection between objects and “things” in the (real) world. There seemed to be agreement that this setup was a useful approach to modeling provenance. We proceeded by reviewing the notions of *specialization-of* and *alternate-of*. These are relations between objects, but the relations are defined such that they can only hold between objects that refer to the same “thing”. Here we observed that some clauses in the definition, related to attribute values that must agree, were redundant. A general critique that was raised is whether the *thing-of* connection from objects to things should be “cast in stone” or be part of an interpretation that can vary.

In the second session on formal models we discussed another topic, namely, provenance information for database query results in the form of provenance polynomials. More specifically, we looked at the case where queries are not merely positive relational algebra expressions, but full relational algebra expression, involving the difference operator. The discussion on including the difference operator was initiated by Floris Geerts’ talk on recording provenance for the SPARQL language where the semantics of one of the SPARQL operators (namely optional) is expressed by means of a “minus” operation. When a formal “minus” operator is added to the provenance polynomial semiring, extended provenance polynomials can be derived that involve the minus operator. Note that to capture that a tuple is *not* in the query result we assign to it provenance “0”. We worked out an example of a difference operation on relations annotated with tuple ids. For example, suppose we compute the expression $R - (R - S)$ on relations where R contains b with tuple-id x_2 and S contains b with tuple-id x_4 . Then the final result will contain b with tuple-id $x_2 - (x_2 - x_4)$. Now Floris points out that if you provide the semiring with additional axioms that imply that $x_2 - (x_2 - x_4) = x_2x_4$, then we get the same final annotation as we would get when computing $R \cap S$, and indeed $R - (R - S)$ is equivalent to $R \cap S$. So, it seems that the full relational algebra with difference can indeed be handled by an extension of the semiring provenance approach. Unfortunately, there are only two papers on handling difference with provenance semirings [1, 2], and these papers do not seem to make very explicit how this can work. Floris Geerts in the end raised some doubts on the axiom $x_2 - (x_2 - x_4) = x_2x_4$, perhaps this is a reason why it is not explicit in the literature.

References

- 1 Y. Amsterdamer, D. Deutch, V. Tannen: On the Limitations of Provenance for Queries With Difference. CoRR abs/1105.2255: (2011)
- 2 F. Geerts, A. Poggi: On database query languages for K-relations. J. Applied Logic 8(2): 173-185 (2010)

5.2 Systems and security perspectives on provenance

Led by Nate Foster, and summarized by Nate Foster (Cornell University, US)

License © © © Creative Commons BY-NC-ND 3.0 Unported license
© Nate Foster

Joint work of All participants of the provenance, systems and security break-out sessions.

We discussed general issues related to provenance and security, as well as some specific security mechanisms provided in systems being developed by two of the participants. Overall, there was broad agreement that security issues are critically important, and that failing to deal with them could hinder the broader adoption of provenance. One clear set of issues concerns the confidentiality and integrity of provenance metadata itself. For example, mechanisms for ensuring that unauthorized users do not access or modify provenance metadata are obviously needed. The group discussed using cryptography as a means for obtaining secure and tamper-proof storage of provenance, but also noted that because provenance tends to be stored for a very long time, current cryptography may not provide sufficient protection. Peter Buneman proposed time-limited archiving systems as a potentially interesting idea for future work. Another set of issues concerns evaluating queries over provenance. It is well known that queries can be used to indirectly obtain information about the underlying data – cf. the case involving the Netflix Prize data [1]. This is exacerbated in systems with provenance, since knowing how a query result was computed can provide useful information to an attacker. The group discussed several scenarios including employee reviews (where provenance might identify the co-workers involved in producing the reviews) and elections (where provenance might reveal an individual’s vote). Although existing work on database privacy seems to provide the basic framework for reasoning about privacy-preserving queries, no systems we know of handle the complicated graph structures often used to represent provenance or adequately captures the “entanglement” between provenance and the underlying data. Lastly, the group discussed whether security mechanisms should be built into the systems that collect provenance or imposed after the fact. Adriane Chapman and Ashish Gehani described the treatment of provenance in PLUS and SPADE. Both systems provide mechanisms for restricting the information incorporated into provenance artifacts.

References

- 1 A. Narayanan, V. Shmatikov. Robust De-anonymization of Large Sparse Datasets. In IEEE Symposium on Security and Privacy (Oakland) 2008, p. 111–125.

5.3 Social Aspects of Provenance

Led by Carole Goble, and summarized by Adriane Chapman (MITRE – McLean, US)

License © © © Creative Commons BY-NC-ND 3.0 Unported license
© Adriane Chapman

Joint work of All participants of the social aspects of provenance break-out sessions.

We discussed the social needs, benefits, risks, obstacles, incentives and challenges of provenance capture and usage. It was noted that there are rewards and incentives for using provenance, which are often reaped by different individuals than the ones who have the burden of reporting provenance. Three key use cases were presented to facilitate discussion:

1. Employee Feedback [1]. Consider three employees give private feedback on a co-worker’s performance. They are willing to do so because their responses are kept private. The

employer’s provenance record could divulge sensitive information about the reviews, e.g. All of the reviews were negative. If the provenance record contains 3 reviews, and there are only 3 other co-workers, the employee knows that all co-workers shared negative feedback.

2. Corporate Structure [2]. Consider an organization with a specific task. Division of labor means that different individuals within the organization have very different jobs. As a reduced example, Alice reads newspapers and synthesizes a report. Bob builds a program to fuse all of Alice’s reports on a given topic. Cathy takes these fused reports, needs to understand the sources originally used (were they trustworthy, is there duplication) and makes a decision (e.g., to invest or not). Doug, the manager, needs to understand how well Alice, Bob and Cathy are performing. Cathy and Doug are obvious users of provenance, but the burden of creation lies more heavily on Alice and Bob.
3. Scientific Usage [3]. A scientific user has the incentive to wish to track provenance for very positive reasons: to enable understanding of scientific results; to receive due credit; etc. However, divulging provenance also has potential negative consequences: someone stealing the secret sauce; someone seeing all of the ugly dead-ends explored; etc.

Using these use cases as a basis, the group explored trade-offs of trust, levels of friendliness, and cost in terms of capturing and exposing all, some or no provenance.

References

- 1 U. Braun, A. Shinnar, and M. Seltzer. Securing Provenance. In USENIX HotSec, 2008.
- 2 A. Chapman and A. Rosenthal. Provenance Needs Incentives for Everyone. In TaPP, 2011.
- 3 C. Goble, D. De Roure, and S. Bechhofer. Accelerating scientists’ knowledge turns. In IC3K, 2012.

5.4 Additional discussions

The participants also held a number of informal research discussions as is normal for a Dagstuhl seminar. Of particular note:

- discussion of semantics and other features of the W3C PROV standard among WG members (Moreau, Groth, Missier, Cheney, Garijo, Eckert, Hartig, Zhao)
- development of a “best practice” mapping from Dublin Core to PROV by Garijo and Eckert.
- discussion of the provenance semiring model among researchers who had not previously been exposed to it, leading to an accessible, informal “nano-tutorial” due to Lois Decambre (lightly edited)

In order to interpret the most informative version of T.J. Green’s provenance polynomials with relational queries that involve only select, project, join/cross product, and union, just imagine that every tuple in a relational database is identified by a unique symbol. You can think of it like a label that is assigned to each tuple. And imagine that these labels are: a, b, c , etc.

Then a provenance polynomial such as $a^2 + 2ab + c^2$ associated with a given tuple (call it x) in the query answer, tells us that x is in the query answer because: the appearance of the tuple a – twice – resulted in x . (That is, some table was joined with itself and the tuple labeled a joined with itself and produced x .)

Or (because the $+$ symbol means “or” ... or “UNION”) the presence of tuple a and b (together) – twice – produced x . So, the tuple a joined with tuple b – in two

different situations – and they both produced x . We know that it happened in two different situations because of the multiplier of 2. We can think of it as a joined with b and then elsewhere in the query processing, a joined with b another time: $2ab = ab + ab$.

Or: the presence of tuple b – twice – results in x . Once again – this represents a self-join where the tuple b joining with itself – produced x .

In the polynomial, multiplication means that both input tuples needed to be present (typically through a join) and addition means that either of the two combinations would be sufficient to produce the output (typically through a union).

So, the provenance polynomial simply tells us, precisely, all of the ways that input tuples combined to produce this output tuple (x , in my example). It's a complete recording of the provenance (or at least, as complete as one can have using semiring annotations); there's no other combination of tuples in the input that could lead to x . I also mentioned that the polynomials (like $a^2 + ab$) are actually combining tuple labels; they are **not** necessarily numerical variables in the classical sense – like one would see in a polynomial in an algebra class in middle school.

I may have also mentioned that one of the ways you can use the polynomial is to figure out if x (in this example) still belongs in the query answer if one or more of the tuples in the input database disappear (or is not trusted or whatever). If the tuple symbol is replaced with '1' if it exists and '0' if it doesn't, then you can find out whether x still belongs in the query answer by evaluating the polynomial.

6 Open Problems

Over the course of their discussions, the seminar participants have identified the following core set of open problems that require investigation.

6.1 Problems related to formal provenance models

Reported by Umut Acar, Sarah Cohen-Boulakia, Paul Groth, Lois Delcambre, Bertram Ludäscher, Simon Miles, Paolo Missier, and Stijn Vansummeren, summarizing discussion by other participants

6.1.1 The semiring based approach towards provenance

For query languages with limited expressive power, like the positive fragment of the relational algebra, recent research has shown it possible to define the formal mathematical structure that provenance annotations should take in order to be able to interpret these annotations in a way that corresponds to the execution of the query. In particular, for the positive fragment of the relational algebra, this mathematical structure is the semiring [10]. Extending this approach to more powerful query languages, such as query languages with aggregation [3] or non-monotonic operators [11, 12, 4] is challenging. Indeed, when considering the difference operator there are various reasonable but non-equivalent mathematical structures that can describe its execution and semantics, all depending on the context in which the provenance annotations are to be used [11, 4].

More work on extending the semiring approach towards provenance with set difference and other non-monotonic operators seems necessary to get a full understanding of the issues involved. In particular:

- Semiring-based approaches do two things: they extend the data model in order to cope with set, bag, probabilistic, etc. kind of data; and they allow for the modeling of annotations. What if we only stick to one single semantics (say set, or bags)? Does difference then still cause a problem?
- How can one redefine the semantics of non-monotonic operators in query languages that operate under an open-world assumption, such as SPARQL, in order to allow for a simple characterization of the structure of provenance annotations?
- What is the limit of the provenance polynomials approach towards provenance? For example, can it be reconciled with approaches such as where-provenance that do not necessarily respect all query equivalences?

6.1.2 Program Traces

In contrast to database query languages with limited expressiveness, it is more difficult to give a detailed provenance account of the execution of a program written in a fully-expressive programming language.

There is a large space of different possible forms of execution traces aimed at different applications:

- profiling, debugging, slicing [14]
- dynamic information flow security [13]
- incremental recomputation (self-adjusting computation [2])
- possibly others (for example bidirectional computation [5] or blame/contracts [9])

Models of provenance used in databases address a special class of computations (and changes that “subtract information”) which make it possible to obtain nice properties, such as the homomorphism commutation property in the semiring model. However, such techniques are relatively fragile with respect to extensions: for example to handle negation, we need to generalize the semiring model in one direction, to handle aggregation, we need to generalize it in another direction, etc. This is analogous to the problems of denotational semantics in classical programming language theory, while modern language researchers often use operational techniques that are easier to combine but arguably more ad hoc. Thus, models of provenance can be developed based on an operational notion of trace which simply records everything that (seemingly) makes sense to record during execution, in a form that can be processed later.

The main challenges for using detailed traces as provenance are:

- Recording control-flow and both control and data dependence relationships linking parts of the input to parts of the program or output in a clean way.
- Defining principled forms of slicing or transformations on traces.
- Identifying good tradeoffs between performance and precision, for example through abstraction or slicing on traces.
- Developing provenance models suitable for high-level explanation for non-technical users, for example users of scientific programming languages such as R.

6.1.3 Provenance in Scientific Workflows

In database queries or when using program slicing, computations may be statically analyzed (at least partially) and thus can be seen as “white box” operations (i.e., one can “see” inside

of them and analyze the operations). Scientific workflows often correspond to “grey boxes”, having some parts that can be seen and analyzed (e.g., the workflow structure or “wiring” itself), but also many other parts that are “black boxes”, i.e., existing third-party services or applications whose code and inner workings are often unknown. On the other hand, scientific workflow systems provide a controlled execution environment with various opportunities to capture detailed provenance information at runtime. The workflow execution models of systems differ widely however, leading to challenges when trying to interpret or interoperate workflow provenance information. Some related questions include:

- Is there a common core that underlies different models of computation across workflow systems and scripting languages?
- Can we enhance runtime provenance recorded by workflow systems with compile-time knowledge about the given model of computation and provenance (cf. [6])?
- In what sense does one model give “more detailed” provenance than another model? And can we find meaningful, formal mappings between different models?
- Is it possible to discern, given a provenance trace, whether the trace could have been produced by one variant of a workflow, but not another (e.g., see [8]), or by one workflow system (e.g., Kepler) but not another (e.g., Taverna)?
- More generally, can we formalize the (provenance) semantics of different workflow systems, define key properties such as “reproducibility” or “replayability”, and prove that given systems do or do not enjoy them?
- Can traces or semantics for concurrent languages be adapted to support modeling and reasoning about provenance?

References

- 1 U. A. Acar, A. Ahmed, J. Cheney and R. Perera. A core calculus for provenance. In *POST*, pages 410–429. Springer-Verlag, 2012.
- 2 U. A. Acar, G. E. Blelloch, and R. Harper. Adaptive functional programming. *ACM Trans. Program. Lang. Syst.*, 28(6):990–1034, 2006.
- 3 Y. Amsterdamer, D. Deutch, V. Tannen: Provenance for aggregate queries. *PODS 2011*: p. 153–164.
- 4 Y. Amsterdamer, D. Deutch, V. Tannen: On the Limitations of Provenance for Queries With Difference. *TAPP 2011*.
- 5 A. Bohannon, J. N. Foster, B. C. Pierce, A. Pilkiewicz, and A. Schmitt. Boomerang: resourceful lenses for string data. In *POPL*, pages 407–419. ACM, 2008.
- 6 S. Bowers, T. McPhillips, and B. Ludäscher. Declarative rules for inferring fine-grained data provenance from scientific workflow execution traces. In *Intl. Provenance and Annotation Workshop (IPAW)*, Santa Barbara, 2012.
- 7 J. Cheney, S. Chong, N. Foster, M. Seltzer, and S. Vansummeren. Provenance: A future history. In *OOPSLA Companion (Onward! 2009)*, pages 957–964, 2009.
- 8 S. Dey, S. Köhler, S. Bowers, and B. Ludäscher. Datalog as a lingua franca for provenance querying and reasoning. In *Workshop on the Theory and Practice of Provenance (TaPP)*, Boston, 2012.
- 9 C. Dimoulas, R. B. Findler, C. Flanagan, and M. Felleisen. Correct blame for contracts: no more scapegoating. In *POPL*, pages 215–226, New York, NY, USA, 2011. ACM.
- 10 T. J. Green, G. Karvounarakis, V. Tannen: Provenance semirings. *PODS 2007*: p. 31–40.
- 11 F. Geerts, A. Poggi: On database query languages for K-relations. *J. Applied Logic* 8(2): p. 173–185 (2010)
- 12 T. J. Green, Z. G. Ives, V. Tannen: Reconcilable Differences. *Theory Comput. Syst.* 49(2): p. 460–488 (2011).

- 13 P. Shroff, S. F. Smith, and M. Thober, “Dynamic dependency monitoring to secure information flow,” in *CSF*. IEEE, 2007.
- 14 M. Weiser. Program slicing. In *ICSE*, pages 439–449, 1981.

6.2 Provenance, security, and confidentiality

Reported by Adriane Chapman, Ashish Gehani, Andrew Martin, and Steve Zdancewic, summarizing discussion by other participants

The discussions of provenance and security covered a number of sub-topics, each with open problems, including confidentiality, integrity, completeness, threat models, and regulation.

6.2.1 Threat models and formalization

Many researchers (including several workshop participants) are developing mechanisms for securing provenance in different systems (e.g., [2, 6, 10]). Sometimes, these mechanisms are straightforward adaptations of standard protection mechanisms (cryptography, digital signatures) to provenance viewed as data. Often, however, the nature of the provenance information makes additional attacks possible — which we may call *provenance-specific attacks* or *provenance failures* [7]. For example, knowing that a particular graph is provenance generated by a known workflow may enable inferences that allow an attacker to guess parts of the graph that were redacted. Definitions of key security properties such as disclosure and obfuscation [1] or privacy for workflow provenance [5] provide a foundation for understanding provenance-specific attacks, on which we can build provably correct policies or mechanisms for securing provenance in different settings (for example for general-purpose programming languages [1]). However, there is currently little recognition of these problems in the formal security world ([1], the first paper on formal foundations for provenance security, appeared only last year) and thus there is little interaction between theory and practice of provenance security.

Moreover, there is currently little work on threat models for provenance, that is, identifying what we believe an attacker can or cannot do and what we want to prevent them from doing. Again, a key issue is identifying how provenance-specific attacks differ from generic attacks on systems or protocols that may happen to involve provenance. Specifically, work on information flow, auditing, integrity, and tracing is relevant, as is work on provenance in concurrency models.

Open problems:

- connecting the practical provenance security mechanisms being deployed in systems with the foundational notions of correctness or security for provenance,
- developing threat models for provenance,
- identifying aspects that make provenance security different from simply securing the underlying data.

6.2.2 Confidentiality

Consider the problem of protecting patient data including its provenance. Naturally, the raw data and provenance can be protected using standard access control or privacy/anonymity techniques (the latter is, of course, already a very hard problem). However, when provenance is also involved, we need to ask why provenance protection is different from the standard problems of protecting confidential data.

For the common case where the provenance is represented as a graph, access control policies on nodes and edges can be established, that limits access to the base information [3, 14]. However, knowing constraints on the structure of the underlying graph (for example knowing that a graph was generated by a known workflow) can make it possible for attackers to infer more information. Similarly, anonymization techniques for graph data in social networks suggests that knowledge of the graph structure can weaken security [1, 4, 11, 12, 16, 17]. However, we cannot assume that the constraints on provenance graphs are secret. Thus, there is a basic tradeoff between confidentiality and utility of provenance.

Open problems:

- Adapting notions of disclosure and obfuscation to provenance graphs
- Understanding the common constraints on provenance graphs and developing policy languages that express common confidentiality requirements
- Identifying limits on safe release of provenance information

6.2.3 Integrity and Completeness

The more value perceived about the data (and its provenance), the greater the motivation for attack. It is not worth protecting a 99 cent piece of data with a \$99 protection strategy. Broadly, integrity has several facets: protecting information from alteration by unauthorized users, being able to prove that information is valid (e.g. has not been changed since creation by an authorized user), and being confident that the information is complete (or at least, that you know how complete it is), for example to detect when changes to source data invalidate other data.

For the first problem, existing techniques such as digital signatures or trusted hardware modules (TPM) may help, as with protecting the integrity of ordinary data. For provenance, it may be more important to provide verifiable links between versions of the same data [10]. In some settings, write-once, read-many (WORM) storage offers a capability to record data that is provably unchanged over time (available as a commodity product).

For the second, being able strongly to tie a (certain version of) the software (and contextual libraries, etc.) to a particular data item apparently generated by that software, is desirable. Techniques of watermarking achieve this well in certain contexts; an approach using the “chain of trust” associated with the TPM is also an active research area [13]. In addition, watermarking has been applied to provenance for video data [7] and sensor data [1].

For the third, the issue of completeness of provenance, a motivating example is a researcher who is discovered to have falsified some results (e.g. the South Korean cloning researcher case a few years ago). Other researchers may have used these results or raw data and now this work needs to be revisited as well. This issue crosses over to the social aspects of provenance surrounding what are you providing, what are the risks, and the benefits. It is also related to the discussion of formal models of provenance (e.g. completeness of traces, reproducibility).

Open problems:

- How can digital signatures, TPMs, WORM storage or other basic mechanisms be combined to ensure provenance is protected from unauthorized alteration? Is it just a matter of protecting the provenance “as data” or is further work needed for different forms of provenance?
- How can we provably certify (or audit) provenance? Can standard watermarking or steganography techniques be used or are new techniques needed? How do the incentives and capabilities to falsify provenance differ from those of ordinary data?
- For the purposes of security, what are appropriate definitions of completeness for provenance?

6.2.4 Regulation

There are many, and often conflicting, laws and regulations regarding provenance. In some cases, the law is specifically concerned with protection of citizens/patients, such as HIPAA and the European Data Protection Directive. These regulations encourage not keeping any, or very little, provenance information because it increases the risk of exposure and attack. On the other hand, some laws, such as Sarbanes-Oxley, facilitate attacks because they require organizations to keep everything.

Open questions:

- How can we ensure that provenance security models and mechanisms are appropriate fits for legal regulations?
- Can provenance techniques provide legally admissible evidence that regulations have or have not been met?

References

- 1 L. Backstrom, C. Dwork, and J. Kleinberg: Wherefore Art Thou R3579X? Anonymized Social Networks, Hidden Patterns, and Structural Steganography, WWW, 2007.
- 2 B. Blaustein, A. Chapman, L. Seligman, M. D. Allen, and A. Rosenthal: Surrogate Parenthood: Protected and Informative Graphs, PVLDB, 2010.
- 3 A. Chebotko, S. Chang, S. Lu, F. Fotouhi, and P. Yang: Scientific Workflow Provenance Querying with Security Views, WAIM, 2008.
- 4 G. Cormode, D. Srivastava, S. Bhagat, and B. Krishnamurthy: Class-based graph anonymization for social network data, PVLDB, vol. 2, 2009.
- 5 S. Davidson, S. Khanna, T. Milo, D. Panigrahi, and S. Roy: Provenance Views for Module Privacy, PODS, 2011.
- 6 A. Gehani and U. Lindqvist, Bonsai: Balanced Lineage Authentication, 23rd Annual Computer Security Applications Conference (ACSAC), IEEE Computer Society, 2007.
- 7 A. Gehani and U. Lindqvist, VEIL: A System for Certifying Video Provenance, 9th IEEE International Symposium on Multimedia (ISM), 2007.
- 8 A. Gehani, B. Baig, S. Mahmood, D. Tariq, and F. Zaffar: Fine-Grained Tracking of Grid Infections, 11th ACM/IEEE International Conference on Grid Computing (GRID), 2010.
- 9 A. Gehani and M. Kim, Mendel: Efficiently Verifying the Lineage of Data Modified in Multiple Trust Domains, 19th ACM International Symposium on High Performance Distributed Computing (HPDC), 2010.
- 10 R. Hasan, R. Sion, M. Winslett, The Case of the Fake Picasso: Preventing History Forgery with Secure Provenance, 7th USENIX Conf. on File and Storage Technologies (FAST), 2009.
- 11 M. Hay, G. Miklau, D. Jensen, D. Towsley, and P. Weis: Resisting Structural Identification in Anonymized Social Networks, VLDB, 2008.
- 12 K. Liu and E. Terzi, Towards Identity Anonymization on Graphs, SIGMOD, 2008.
- 13 J. Lyle and A. Martin, Trusted Computing and Provenance: Better Together, in proceedings TaPP'10, USENIX.
- 14 A. Rosenthal, L. Seligman, A. Chapman, and B. Blaustein: Scalable Access Controls for Lineage, in Theory and Practice of Provenance, 2009.
- 15 J. Zhang, A. Chapman, and K. LeFevre: Fine-Grained Tamper-Evident Data Pedigree, Secure Data Management, 2009.
- 16 E. Zheleva and L. Getoor: Preserving the Privacy of Sensitive Relationships in Graph Data, PinKDD, 2007.
- 17 B. Zhou and J. Pei: Preserving Privacy in Social Networks Against Neighborhood Attacks, ICDE, 2008.

6.3 Social Aspects of Provenance

Reported by Carole Goble and Jim Frew, summarizing discussions involving Shawn Bowers, Kai Eckert, Paul Groth, Luc Moreau, Perdita Stevens, Jun Zhao, and other participants

Provenance discussions have typically been couched in terms of benefit to the consumer. Anecdotally, users are enthusiastic about provenance becoming available to them but less obliging about supplying provenance on their data to others. At the seminar, a discussion group covered a wide range of issues concerning the rewards, risks, burdens, and benefits of provenance; how these relate to technical requirements or proposals; and how to evaluate whether current or future solutions address these needs (and are worth the costs).

The example of traceability in software engineering gives cause for concern: despite a large amount of research on the subject, experience in the field suggests that the benefits of adopting traceability techniques may not outweigh their costs.

The discussion group produced a substantial outline which (together with other materials in this report and on the seminar wiki) may form the basis for a longer “manifesto” paper by participants in the seminar. The following discussion of open problems is distilled from that outline.

6.3.1 Rewards, risks, burden, benefit of provenance

Part of the problem of identifying the rewards, risks, burdens, and benefits of provenance is terminological: people disagree on what provenance is, and whether it “is” metadata, trust, quality or identity information, or just a record of this information. The working group identified the different needs and goals of provenance consumers and producers.

- How can we untangle confusion among provenance, metadata, trust, quality, and identity?
- How can we develop infrastructure that provides “stealth/ninja provenance” – merging into existing information infrastructure.
- How can we design appropriate provenance capture mechanisms based on (clear understanding of) what we, or users, will eventually want to use it for?

6.3.2 Technical requirements and capabilities

The group also identified a lifecycle for provenance production: capture, preparation, sharing, and using, and identified benefits and risks for producers and consumers of provenance.

- How can we design mechanisms that take into account the motivations (and demotivations) on provenance producers (voluntary, peer pressure, mandatory) and different classes of consumers (self, friends, family/colleagues, public)?
- Likewise, how can we develop systems that take into account the different stages of production (raw data, preliminary results, polished results, publication)?
- How do we reconstruct or complete provenance when it was not originally captured?

6.3.3 Evaluation

Finally, the group produced a draft checklist for projects or tool providers to characterize what aspects of provenance they do or do not handle. This could serve as a basis for comparison of different techniques, offsetting economic costs considerations.

- How can we evaluate compliance with a collection of requirements on provenance systems?

- How would the costs/benefits of provenance be affected by developing standards or infrastructure that provides it pervasively, rather than in heterogeneous ways in different systems? Is it worth it?

6.3.4 Pointers to the literature

Much of the discussion can be framed by the literature in data sharing and collaboration behaviours in knowledge enterprises and scientific communities [9, 11, 8, 3, 6, 1, 5]. There is also a useful literature (partially covered by the above) examining the incentives, behaviour patterns, models and quality of voluntary information. Additional references include [2, 10]. The whole area of motivations to contribute wikis is a useful area to look at (e.g. [7]). Jane Hunter had previously highlighted sensitivities around the publishing of provenance and a desire to “provenance spring clean” [4].

References

- 1 Borgman, C.L. The Conundrum of Sharing Research. *Journal of the American Society for Information Science and Technology*, 1–40 (2011)
- 2 Andrew J. Flanagan and Miriam J. Metzger, The credibility of volunteered geographic information. *GeoJournal* (2008) 72:137–148
- 3 Howison, J., Herbsleb, J.D. Scientific software production: incentives and collaboration. *Proc ACM 2011 Conf Computer Supported Cooperative Work*, 513–522 (2011)
- 4 Hunter, J. Scientific Publication Packages – A Selective Approach to the Communication and Archival of Scientific Output., *Intl J of Digital Curation* 1 (1) (2006)
- 5 Liebowitz, J., Ayyavoo, N., Nguyen, H., Carran, D., Simien, J. Cross-generational knowledge flows in edge organizations. *Industrial Management & Data Systems*, 107(8) 1123–1153 (2007)
- 6 Nielson, M. *Reinventing Discovery: The New Era of Networked Science*. Princeton University Press (2011)
- 7 Stacey Kuznetsov. 2006. Motivations of contributors to Wikipedia. *SIGCAS Comput. Soc.* 36, 2, Article 1 (June 2006). See also: <http://www.staceyk.org/personal/WikipediaMotivations.pdf>
- 8 Stodden, V. The Scientific Method in Practice: Reproducibility in the Computational Sciences. MIT Sloan Research Paper No. 4773-10. doi:10.2139/ssrn.1550193 (2010)
- 9 Tenopir C, Allard S, Douglass K, Aydinoglu AU, Wu L, et al. (2011) Data Sharing by Scientists: Practices and Perceptions. *PLoS ONE* 6(6): e21101. doi:10.1371/journal.pone.0021101
- 10 Wikipatterns.com: a toolbox of patterns & anti-patterns, and a guide to the stages of wiki adoption. <http://www.wikipatterns.com/display/wikipatterns/Wikipatterns>
- 11 Yakowitz, J. Tragedy of the Data Commons. *Harvard J of Law and Tech*, Vol. 25 (2011)

Participants

- Umut A. Acar
MPI for Software Systems –
Kaiserslautern, DE
- Shawn Bowers
Gonzaga Univ. – Spokane, US
- Peter Buneman
University of Edinburgh, GB
- Adriane Chapman
MITRE – McLean, US
- James Cheney
University of Edinburgh, GB
- Stephen Chong
Harvard University, US
- Sarah Cohen-Boulakia
Université Paris Sud, FR
- Victor Cuevas-Vicenttin
St. Martin-d’Heres, FR
- Lois Delcambre
Portland State University, US
- Kai Eckert
Universität Mannheim, DE
- Nate Foster
Cornell University, US
- Juliana Freire
Polytechnic Institute of NYU –
Brooklyn, US
- James Frew
University of California – Santa
Barbara, US
- Irimi Fundulaki
FORTH – Heraklion, GR
- Daniel Garijo
Universidad Politécnica de
Madrid, ES
- Floris Geerts
University of Edinburgh, GB
- Ashish Gehani
SRI – Menlo Park, US
- Carole Goble
University of Manchester, GB
- Todd J. Green
University of California – Davis
and LogicBlox, US
- Paul Groth
Free Univ. – Amsterdam, NL
- Torsten Grust
Universität Tübingen, DE
- Olaf Hartig
Humboldt Univ. zu Berlin, DE
- Melanie Herschel
Université Paris Sud, FR
- Bertram Ludaescher
Univ. of California – Davis, US
- Andrew Martin
University of Oxford, GB
- Simon Miles
King’s College – London, GB
- Paolo Missier
Newcastle University, GB
- Luc Moreau
University of Southampton, GB
- Leon J. Osterweil
University of Massachusetts –
Amherst, US
- Christopher Re
University of Wisconsin –
Madison, US
- Vladimiro Sassone
University of Southampton, GB
- Martin Schäler
Universität Magdeburg, DE
- Margo Seltzer
Harvard University, US
- Christian Skalka
University of Vermont, US
- Perdita Stevens
University of Edinburgh, GB
- Wang-Chiew Tan
IBM Research & University of
California – Santa Cruz, US
- Jan Van den Bussche
Hasselt University, BE
- Stijn Vansummeren
Université Libre de Bruxelles, BE
- Marianne Winslett
Univ. of Illinois – Urbana, US
- Steve Zdancewic
University of Pennsylvania, US
- Jun Zhao
University of Oxford, GB

