SIG Proceedings Paper in LaTeX Format*

Extended Abstract†

Ben Trovato‡
Institute for Clarity in Documentation
P.O. Box 1212
Dublin, Ohio 43017-6221
trovato@corporation.com

G.K.M. Tobin§
Institute for Clarity in Documentation
P.O. Box 1212
Dublin, Ohio 43017-6221
webmaster@marysville-ohio.com

Lars Thørväld¶
The Thørväld Group
1 Thørväld Circle
Hekla, Iceland
larst@affiliation.org

Lawrence P. Leipuner
Brookhaven Laboratories
P.O. Box 5000
lleipuner@researchlabs.org

Sean Fogarty
NASA Ames Research Center
Moffett Field, California 94035
fogartys@amesres.org

Charles Palmer
Palmer Research Laboratories
8600 Datapoint Drive
San Antonio, Texas 78229
cpalmer@prl.com

John Smith
The Thørväld Group
jsmith@affiliation.org

Julius P. Kumquat
The Kumquat Consortium
jpkumquat@consortium.net

Figure 1: This is a teaser

ABSTRACT

This paper provides a sample of a LaTeX document which conforms, somewhat loosely, to the formatting guidelines for ACM SIG Proceedings.

*Produces the permission block, and copyright information
†The full version of the author's guide is available as `acmart.pdf` document
‡Dr. Trovato insisted his name be first.
§The secretary disavows any knowledge of this author's actions.
¶This author is the one who did all the really hard work.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
WOODSTOCK'97, El Paso, Texas USA
© 2016 Copyright held by the owner/author(s). 123-4567-24-567/08/06…$15.00
DOI: 10.475/123_4

CCS CONCEPTS

•**Computer systems organization** →**Embedded systems;** *Redundancy;* Robotics; •**Networks** →Network reliability;

KEYWORDS

ACM proceedings, LaTeX, text tagging

ACM Reference format:
Ben Trovato, G.K.M. Tobin, Lars Thørväld, Lawrence P. Leipuner, Sean Fogarty, Charles Palmer, John Smith, and Julius P. Kumquat. 1997. SIG Proceedings Paper in LaTeX Format. In *Proceedings of ACM Woodstock conference, El Paso, Texas USA, July 1997 (WOODSTOCK'97),* 8 pages.
DOI: 10.475/123_4

1 INTRODUCTION

In recent years,the Internet of Things (IoT) has gathered significant traction which has led to the exponential increase in the number of devices connected to the internet. It has

revolutionized devices to device communication, which in turn optimizes efficiency and improves the standard of living. According to a report released by Cisco, it is estimated that a total of 50 million devices will be connected to the internet by the year 2020. With the vast amounts of connected heterogeneous devices, security and privacy risks are increased. Data provenance is instrumental to the security IoT data. Provenance describes a holistic history of operations performed on a data object from its point of creation. Provenance can be used as a measure to ensure trust and also for forensic analysis in an event of malicious attack. Provenance collection system is of immense benefit to IoT devices however provenance incurs additional storage overhead. Continuous provenance collection has the tendency to generate more data than the data it describes. Work done by Brian et al demonstrates that changes to a source file in the Linux kernel results to two kilobytes of additional provenance data when the kernel is recompiled. Additionally, Figure x displays a graph of the growth of provenance data for a raspberry pi collecting temperature and humidity readings. This sensor reading is stored in CTF format. From the diagram we can see that the data grows rapidly with time. This motivates the need for an efficient pruning technique to reduce storage overhead that might be incurred by including provenance in an IoT application. It is of utmost importance to provide a method of pruning provenance data generated thereby reducing the storage overhead. This requirement is further motivated by the constrained memory and computing power of IoT devices. Pruning can be described as removing provenance data in order to conserve storage.

In this paper, we propose a fine-grained, policy model for provenance pruning. This model provides an expressive policy language which allows device administrators the flexibility of specifying what provenance data to store. We argue that policy is an effective way of addressing the data pruning problem created by automatic provenance collection. With access control, we allow the flexibility of deciding what provenance data is discarded thereby eliminating unwanted provenance information. Provenance sometimes generates information that might be considered uninteresting. For example, a system collecting the provenance of system events such as device internal system state might be considered uninteresting to a researcher working on collecting temperature and humidity readings or a device administrator interested on. We implement a proof of concept system using XACML, a fine-grained attribute based access control policy language which evaluates request based on a policy specifications. XACML serves as the foundation for our framework.We compared our performance of the proposed framework with state of the art on provenance pruning (e.g web compression + dictionary). The remaining portion of the paper is outlined as follows: Section 2 discusses background information on IoT provenance pruning. Section 3 talks about related work on provenance pruning. Section 4 discusses implementation techniques, section 5 talks discusses experimental evaluation and finally, section 6 concludes and discusses future works.

IoT trace is derived in the form of CTF output. This output stream is pruned to reduce storage overhead.

CTF is a binary format that allows dynamic instrumentation trace of a applications written in C/C++. The source code for applications is compiled with the generated c code that barectf creates for the IoT trace event.

barectf is chosen for tracing IoT device because it generates ANCI C code which is lightweight and can fit into most microcontrollers.

2 BACKGROUND

This section describes key concepts of data provenance, IoT characteristics, and provenance models. It also provides motivating example for the need for provenance pruning.

Internet of Things

There is no standard definition for IoT, however, researchers have tried to define the concept of connected ẗhings. The concept of IoT was proposed by Mark Weiser in the early 1990s which represents a way in which the physical objects, ẗhings, can be connected to the digital world. Gubbi et al defines the IoT as an interconnection of sensing and actuating devices that allows data sharing across platforms through a centralized framework. We define (IoT) as follows:

Definition 2.1. The Internet of Things (IoT) is a network of heterogeneous devices with sensing and actuating capabilities communicating over the internet.

The notion of IoT has been attributed to smart devices. The interconnectivity between various heterogeneous devices allows for devices to share information in a unique manner. Analytics is a driving force for IoT. With analytics, devices can learn from user data to make smarter decisions. This notion of smart devices is seen in various commercial applications such as smartwatches, thermostats that automatically learns a user patterns. The ubiquitous nature of these devices make them ideal choices to be included in consumer products. IoT architecture represents a functional hierarchy of how information is disseminated across multiple hierarchies contained in an IoT framework; from devices which contain sensing and actuating capabilities to massive data centers (cloud storage). Knowing how information is transmitted across layers allows a better understanding on how to model the flow of information across actors contained in an IoT hierarchy. Figure 1 displays the IoT architecture and the interactions between the respective layers. IoT architecture

consists of four distinct layers: The sensor and actuator layer, device layer, gateway layer and the cloud layer. The base of the architectural stack consist of sensors and actuators which gathers provenance information and interacts with the device layer. The device layer consists of devices (e.g mobile phones, laptops, smart devices) which are responsible for aggregating data collected from sensors and actuators. These devices in turn forwards the aggregated data to the gateway layer. The gateway layer routes and forwards data collected from the device later. It could also serve as a medium of temporary storage and data processing. The cloud layer is involved with the storage and processing of data collected from the gateway layer. Note that the resource constraints decreases up the architectural stack with the cloud layer having the most resources (memory, power computation) and the sensor- actuator layer having the least.

Automatic provenance collection. For pruning in general, web compression and web compression + dictionary compression are techniques that has been effectively suggested in the literature. This techniques are considered efficient in pruning provenance but they fail to address the issue of pruning provenance data that might be uninteresting to the device administrator of an automatic provenance collection system. Motivated by the work of Brian et al suggests the need for a policy driven provenance collection system in which system events that are important to device administrator is collected thereby leaving out interesting events which increases storage overhead. While we consider pruning to be an effective way to address the issue of storage overhead, there exist a tradeoff between the data that is pruned and its importance to the collective provenance. To this end, we define a probabilistic model that estimates the risk involved in deleting unwanted provenance data. With is information, the user is left to decide if the provenance data in question that is deleted is considered important to the provenance chain.

For provenance to be effective in forensic analysis, it has to be complete

With the recent data explosion [22] due to the large influx in amounts of interconnected devices, information is disseminated at a fast rate and with this increase involves security and privacy concerns. Creating a provenance-aware system is beneficial to IoT because it ensures the trust and integrity of interconnected devices. Enabling provenance collection in IoT devices allows these devices to capture valuable information which enables backtracking in an event of a malicious attack.

Data Provenance

The Oxford English dictionary defines provenance as the place of origin or earliest known history of something". An

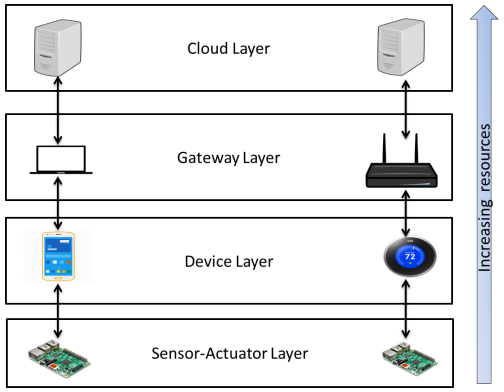

Figure 2: IoT Architecture Diagram. The arrows illustrates the interaction between data at various layers on the architecture.

example of provenance can be seen with a college transcript. A transcript is the provenance of a college degree because it outlines all of the courses satisfied in order to attain the degree. In the field of computing, data provenance, also known as data lineage, can be defined as the history of all activities performed on entities from its creation to its current state. Cheney et al. decribes provenance as the origin and history of data from its lifecycle. Buneman et al describes provenance from a database perspective as the origin of data and the steps in which it is derived in the database system. We formally define provenance as follows:

Definition 2.2. Data provenance of an entity is a comprehensive history of activities that occur on that entity from its creation to its present state.

Provenance ensures trust and integrity of data [**?**]. It outlines causality and dependency between all objects involved in the system and allows for the verification of the source of data. Causality and dependency are used to determine the relationship between multiple objects. The relationship in which provenance denotes can in turn be used in digital forensics [**?**] to investigate the cause of a malicious attack and also in intrusion detection systems to further enhance the security of computing devices.

Provenance Characteristics

Since provenance denotes the who, where and why of data transformation, it is imperative that data disseminated in an IoT architecture satisfies the required conditions. The characteristics of data provenance are outlined in detail below.

- Who: This characteristic provides information on activities made to an entity. Knowing the "who" characteristic is essential because it maps the identity of

modification to a particular data object. An example of "who" in an IoT use case is a sensor device identifier.

- Where: This characteristic denotes location information in which data transformation was made. This provenance characteristic could be optional since not every data modification contains location details.
- When: This characteristic denotes the time information at which data transformation occurred. This is an essential provenance characteristic. Being able to tell the time of a data transformation allows for tracing data irregularities.
- What: This characteristic denotes the transformation is applied on a data object. A use case for IoT can be seen in the various operations (create, read, update, and delete) that could be performed on a file object.

There are two ways of pruning Provenance data: Provenance can be pruned at collection before it is been committed to the disk or after being recorded to disk. Policy defines rules and actions that should be taken if any of the rules applies. Access control in this case is used as a tool for pruning provenance data stored on an IoT device. It can also be extended for traditional access control measures. Data pruning is an essential problem for automatic provenance collection. Observed provenance and disclosed provenance. Observed provenance involves automatic collection of system states and changes. One major drawback of this method is that all system events are provenanced including irrelevant system provenance which incurs more storage overhead. Described provenance on the other hand, allows a user to provide a workflow of how what provenance the system is intended to generate and an engine to execute the workflow described.

3 SYSTEM MODEL

We have already seen several typeface changes in this sample. You can indicate italicized words or phrases in your text with the command `\textit`; emboldening with the command `\textbf` and typewriter-style (for instance, for computer code) with `\texttt`. But remember, you do not have to indicate typestyle changes when such changes are part of the *structural* elements of your article; for instance, the heading of this subsection will be in a sans serif[1] typeface, but that is handled by the document class file. Take care with the use of[2] the curly braces in typeface changes; they mark the beginning and end of the text that is to be in the different typeface.

[1]Another footnote, here. Let's make this a rather short one to see how it looks.
[2]A third, and last, footnote.

You can use whatever symbols, accented characters, or non-English characters you need anywhere in your document; you can find a complete list of what is available in the *LaTeX User's Guide* [7].

Data Model

Provenance data is collected using dynamic instrumentation by attaching barectf trace endpoints to dynamically generate CTF trace. This information is stored on the IoT device. Provenance trace is Sensor and actuator reading is represented as a tuple $< ts, [r_1, r_2, ..., r_n] >$ where ts is denoted as the timestamp and $r_1, ...r_n$ denotes the various data points from the sensor readings. Access control is enforced using a modified version of the security punctuations model.

The policy language is flexible and written in a way that is easily understood. Policy specifies the appropriate groups for each sensor readings. Every sensor attribute is grouped. For each group, there exist and associated role For each role, there is an action of permit or deny. The user who serves as the device administrator is in charge of appropriate sensor data groups. This information is combined to provide permit or deny decisions. This policy with the appropriate request is evaluated to produce appropriate access decision.

Let G be a sensor group and R represents user roles, and U represent

For pruning, access control is used to determine if provenance data should be stored on the device or not this way, data that is discarded is considered invaluable to the organization.

Parts of provenance in which data is deleted can also be marked as deleted theryby maintaining the provenance chain. Data deleted can also be marked. A hybrid apporach which emplys policy and compression can also be implemented.

For example, a research scientist requires temperature/humidity data for research on the environmental outcome over time. He tries to collects temperature and humidity readings of the environment using a raspberry pi and temp/humidity sensor. The sensor readings and relationship between other sensor readings forms the basis of provenanc. provenance is collected from the sensor and stored on the raspberry pi. The device administrator in this case is the scientist performing the experiment. They decide that they only want to store all temperature from a certain time interval. Also, they set a storage threshold, which after this provenance data is automatically discarded.

We also calculate

Access Control Model. A formula that appears in the running text is called an inline or in-text formula. It is produced by the **math** environment, which can be invoked with the usual `\begin . . . \end` construction or with the short form `$ . . . $`. You can use any of the symbols and structures,

from α to ω, available in LaTeX [7]; this section will simply show a few examples of in-text equations in context. Notice how this equation: $\lim_{n\to\infty} x = 0$, set here in in-line math style, looks slightly different when set in display style. (See next section).

Display Equations. A numbered display equation—one set off by vertical space from the text and centered horizontally—is produced by the **equation** environment. An unnumbered display equation is produced by the **displaymath** environment.

Again, in either environment, you can use any of the symbols and structures available in LaTeX; this section will just give a couple of examples of display equations in context. First, consider the equation, shown as an inline equation above:

$$\lim_{n\to\infty} x = 0 \tag{1}$$

Notice how it is formatted somewhat differently in the **displaymath** environment. Now, we'll enter an unnumbered equation:

$$\sum_{i=0}^{\infty} x + 1$$

and follow it with another numbered equation:

$$\sum_{i=0}^{\infty} x_i = \int_0^{\pi+2} f \tag{2}$$

just to demonstrate LaTeX's able handling of numbering.

4 PROPOSED MODEL

The policy framework consists of a policy engine. The policy engine contains authorization and enforcement components that provides and enforce decisions on how provenance data should be stored. A policy document is a component of the policy framework. It identifies provenance data that is considered relevant to the IoT application. Our policy architecture is modeled using the Common Open Policy Service (COPS) Standard [**?**]. COPS consists of components for policy generation, evaluation and enforcement. The Policy Enforcement Point (PEP) enforces decisions received from the Policy Decision Point (PDP). The PDP evaluates policies and generates decision based on the evaluation. The model can be extended to include a secondary decision point (SDP) which allows for distributed policy evaluation, thus freeing up the PDP from communication bottlenecks caused by large amounts of requests received by a single PDP. Figure 3 below illustrates the system architecture of our proposed policy-based storage framework. Different layers of the IoT architecture contain different decision and enforcement components. The sensor-actuator layer of the IoT architecture is omitted because it has negligible memory resources and the sensors and actuators are usually physically part of a device in the device layer and as such does not have any data to prune.

Policy document which is generated by the policy creator serves as an input to the PDP component and is evaluated at the device, gateway and cloud layer. The PEP which is involved with generating requests is located in the device and gateway layer. SDPs can be located in the gateway layer, which allows for policy evaluation without incurring additional network overhead of communicating with the PDP located in the cloud layer.

Using the use case of the smart home depicted in chapter 2, a policy framework could be implemented and incorporated into the IoT architecture which allows a device administrators to specify what kinds of provenance data to collect. The policy acts an enforcement point providing an efficient storage mechanism in a resource constrained environment.

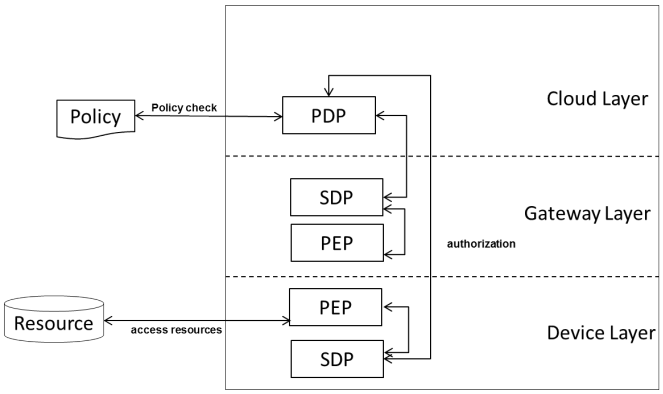

Figure 3: Policy based system architecture which allows for effective storage of provenance data

5 EXPERIMENT

6 DISCUSSION

Citations

Citations to articles [2–4, 6], conference proceedings [4] or maybe books [7, 8] listed in the Bibliography section of your article will occur throughout the text of your article. You should use BibTeX to automatically produce this bibliography; you simply need to insert one of several citation commands with a key of the item cited in the proper location in the `.tex` file [7]. The key is a short reference you invent to uniquely identify each work; in this sample document, the key is the first author's surname and a word from the title. This identifying key is included with each item in the `.bib` file for your article.

The details of the construction of the `.bib` file are beyond the scope of this sample document, but more information

Table 1: Frequency of Special Characters

Non-English or Math	Frequency	Comments
Ø	1 in 1,000	For Swedish names
π	1 in 5	Common in math
\$	4 in 5	Used in business
Ψ_1^2	1 in 40,000	Unexplained usage

can be found in the *Author's Guide*, and exhaustive details in the *LaTeX User's Guide* by Lamport [7].

This article shows only the plainest form of the citation command, using `\cite`.

Tables

Because tables cannot be split across pages, the best placement for them is typically the top of the page nearest their initial cite. To ensure this proper "floating" placement of tables, use the environment **table** to enclose the table's contents and the table caption. The contents of the table itself must go in the **tabular** environment, to be aligned properly in rows and columns, with the desired horizontal and vertical rules. Again, detailed instructions on **tabular** material are found in the *LaTeX User's Guide.*

Immediately following this sentence is the point at which Table 1 is included in the input file; compare the placement of the table here with the table in the printed output of this document.

To set a wider table, which takes up the whole width of the page's live area, use the environment **table*** to enclose the table's contents and the table caption. As with a single-column table, this wide table will "float" to a location deemed more desirable. Immediately following this sentence is the point at which Table 2 is included in the input file; again, it is instructive to compare the placement of the table here with the table in the printed output of this document.

It is strongly recommended to use the package booktabs [5] and follow its main principles of typography with respect to tables:

(1) Never, ever use vertical rules.
(2) Never use double rules.

It is also a good idea not to overuse horizontal rules.

Figures

Like tables, figures cannot be split across pages; the best placement for them is typically the top or the bottom of the page nearest their initial cite. To ensure this proper "floating" placement of figures, use the environment **figure** to enclose the figure and its caption.

This sample document contains examples of `.eps` files to be displayable with LaTeX. If you work with pdfLaTeX, use

Figure 4: A sample black and white graphic.

Figure 5: A sample black and white graphic that has been resized with the `includegraphics` command.

files in the `.pdf` format. Note that most modern TeX systems will convert `.eps` to `.pdf` for you on the fly. More details on each of these are found in the *Author's Guide.*

As was the case with tables, you may want a figure that spans two columns. To do this, and still to ensure proper "floating" placement of tables, use the environment **figure*** to enclose the figure and its caption. And don't forget to end the environment with **figure***, not **figure**!

Theorem-like Constructs

Other common constructs that may occur in your article are the forms for logical constructs like theorems, axioms, corollaries and proofs. ACM uses two types of these constructs: theorem-like and definition-like.

Here is a theorem:

Theorem 6.1. *Let f be continuous on $[a, b]$. If G is an antiderivative for f on $[a, b]$, then*

$$\int_a^b f(t)\,dt = G(b) - G(a).$$

Here is a definition:

Definition 6.2. If z is irrational, then by e^z we mean the unique number that has logarithm z:

$$\log e^z = z.$$

The pre-defined theorem-like constructs are **theorem**, **conjecture**, **proposition**, **lemma** and **corollary**. The pre-defined definition-like constructs are **example** and **definition**. You can add your own constructs using the *amsthm* interface [1]. The styles used in the `\theoremstyle` command are **acmplain** and **acmdefinition**.

Another construct is **proof**, for example,

Table 2: Some Typical Commands

Command	A Number	Comments
`\author`	100	Author
`\table`	300	For tables
`\table*`	400	For wider tables

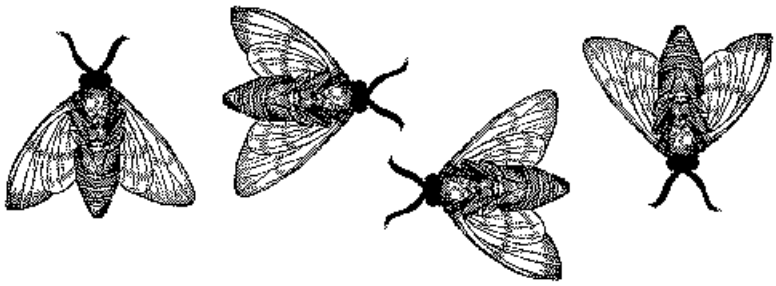

Figure 6: A sample black and white graphic that needs to span two columns of text.

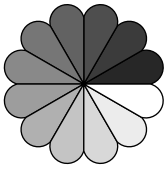

Figure 7: A sample black and white graphic that has been resized with the `includegraphics` command.

Proof. Suppose on the contrary there exists a real number L such that

$$\lim_{x\to\infty} \frac{f(x)}{g(x)} = L.$$

Then

$$l = \lim_{x\to c} f(x) = \lim_{x\to c} \left[gx \cdot \frac{f(x)}{g(x)} \right] = \lim_{x\to c} g(x) \cdot \lim_{x\to c} \frac{f(x)}{g(x)} = 0 \cdot L = 0,$$

which contradicts our assumption that $l \neq 0$. □

7 CONCLUSIONS

Provenance collection offers benefits to IoT. It provides trust of data across various layers of the framework. Provenance collection leads to memory. This ensures flexibility in the amount of provenance data stored on the IoT device. It also offers device administrators authority to decide a limit on the amount of provenance data that can be retained ensuring storage efficiency. In this paper, we present a policy-based approach to provenance data pruning.

A HEADINGS IN APPENDICES

The rules about hierarchical headings discussed above for the body of the article are different in the appendices. In the **appendix** environment, the command **section** is used to indicate the start of each Appendix, with alphabetic order designation (i.e., the first is A, the second B, etc.) and a title (if you include one). So, if you need hierarchical structure *within* an Appendix, start with **subsection** as the highest level. Here is an outline of the body of this document in Appendix-appropriate form:

Introduction

The Body of the Paper

Type Changes and Special Characters.

Math Equations.

Inline (In-text) Equations.

Display Equations.

Citations.

Tables.

Figures.

Theorem-like Constructs.

A Caveat for the TEX Expert.

Conclusions

References

Generated by bibtex from your `.bib` file. Run latex, then bibtex, then latex twice (to resolve references) to create the `.bbl` file. Insert that `.bbl` file into the `.tex` source file and comment out the command `\thebibliography`.

B MORE HELP FOR THE HARDY

Of course, reading the source code is always useful. The file `acmart.pdf` contains both the user guide and the commented code.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Yuhua Li for providing the matlab code of the *BEPS* method.

The authors would also like to thank the anonymous referees for their valuable comments and helpful suggestions. The work is supported by the National Natural Science Foundation of China under Grant No.: 61273304 and Young Scientsts' Support Program (http://www.nnsf.cn/youngscientsts).

REFERENCES

[1] American Mathematical Society 2015. *Using the amsthm Package.* American Mathematical Society. http://www.ctan.org/pkg/amsthm.

[2] Mic Bowman, Saumya K. Debray, and Larry L. Peterson. 1993. Reasoning About Naming Systems. *ACM Trans. Program. Lang. Syst.* 15, 5 (November 1993), 795–825. DOI: http://dx.doi.org/10.1145/161468.161471

[3] Johannes Braams. 1991. Babel, a Multilingual Style-Option System for Use with LaTeX's Standard Document Styles. *TUGboat* 12, 2 (June 1991), 291–301.

[4] Malcolm Clark. 1991. Post Congress Tristesse. In *TeX90 Conference Proceedings.* TeX Users Group, 84–89.

[5] Simon Fear. 2005. *Publication quality tables in LaTeX.* http://www.ctan.org/pkg/booktabs.

[6] Maurice Herlihy. 1993. A Methodology for Implementing Highly Concurrent Data Objects. *ACM Trans. Program. Lang. Syst.* 15, 5 (November 1993), 745–770. DOI: http://dx.doi.org/10.1145/161468.161469

[7] Leslie Lamport. 1986. *LaTeX User's Guide and Document Reference Manual.* Addison-Wesley Publishing Company, Reading, Massachusetts.

[8] S.L. Salas and Einar Hille. 1978. *Calculus: One and Several Variable.* John Wiley and Sons, New York.