2016

A Review of the Literature on Configuration Management Tools

Johannes Hintsch
Otto von Guericke Universitat, Magdeburg, johannes.hintsch@ovgu.de

Carsten Görling
Otto von Guericke University Magdeburg, carsten.goerling@ovgu.de

Klaus Turowski
Otto von Guericke University Magdeburg, klaus.turowski@ovgu.de

6. A Review of the Literature on Configuration Management Tools

Johannes Hintsch, Carsten Görling, Klaus Turowski
Otto von Guericke University Magdeburg
{johannes.hintsch|carsten.goerling|klaus.turowski}@ovgu.de

Abstract

Configuration management tools help administrators in defining and automating system configurations. With cloud computing, host numbers are likely to grow. IaaS (infrastructure as a service) offerings with pay-per-use pricing models make fast and effective deployment of applications necessary. Configuration management tools address both challenges. In this paper, the existing research on this topic is reviewed comprehensively. Readers are provided with a descriptive analysis of the published literature as well as with an analysis of the content of the respective research works. The paper serves as an overview for researchers who are new to the topic. Furthermore, it serves to identify work related to an intended research field and identifies research gaps. Practitioners are provided with a means to identify solutions to their organizational problems.

Keywords

Configuration Management Tools, Literature Review, Descriptive Analysis, Content-Oriented Analysis.

1. Introduction

Configuration management (CM) tools play a crucial role in distributed computing concepts such as grid (Fischer et al., 2014) and cloud computing (Wettinger et al., 2013), but also in managing organizations' heterogeneous system landscapes (Schaefer et al., 2013). They automate, standardize, and modularize administration tasks (Delaet et al., 2010). CM tools are of interest to system operations staff (Fischer et al., 2014), but also to researchers (Wettinger et al., 2013), as these tools, for example, are a means to modularize application components in a way that makes reassembling them to more complex applications feasible, even when such application component modules were designed and developed by different organizations (cf. (Meyer et al., 2013)). CM is defined by the IT Infrastructure Library (ITIL) as *"the process responsible for ensuring that the assets required to deliver services are properly controlled, and that accurate and reliable information about those assets is available when and where it is needed. This information includes details of how the assets have been configured and the relationships between assets."* (Rance, 2011, p. 328)

ITIL defines assets as resources and capabilities (Rance, 2011, p. 302) and systems are the resources whose configuration is managed by CM tools (Delaet et al., 2010). ITIL does not give specific recommendations for tool support of the defined processes. However, several proprietary and open source CM tools (Delaet et al., 2010) that implement CM exist.

Besides the mentioned relevance of CM for service operation other influences can be identified. Configuration management, along with their primary purposes of automation and standardization, address IT-related goals (e.g. audibility and controllability) and enterprise-related goals (e.g. risk and service management) by keeping configuration specifications at a central location and by configuring managed nodes as specified.

For advancing knowledge in a domain Webster and Watson (2002) describe the importance of reviewing past research. Therefore, to provide an overview on the state of the art of CM tools, a structured literature review was conducted and is presented in this paper.

A broad tool-oriented search scope is chosen because operational CM has implications for different areas in an organization as described above. This work is valuable to researchers who want to gain an overview of this topic. To practitioners it can serve as a map that assists their search for publications addressing specific problems in their organizations. Valuable to both, research gaps and paths for future work are identified.
The following descriptive and content-oriented research questions are addressed in this paper:

- Which authors have when and where published peer-reviewed papers about this topic?
- Which configuration management tools are referenced most frequently in the literature?
- Which topics have been studied?

The remainder of the paper is structured as follows. It begins with an overview of the research background, which first introduces CM tools and then reports on related work in section 2. In section 3 the research design is presented. The review is presented in section 4 and the findings are discussed in section 5. In the last section 6 an outlook on possible paths of future work is provided.

2. Research Background

In the first subsection, an overview about CM tools is given and in the second subsection related work is briefly discussed.

2.1 Configuration Management Tools

In 2010 Delaet et al. (2010) published a comparative framework for configuration management tools and defined the tools' architectural essence. Every tool provides the administrator with an interface, which is used to specify the configuration of the managed devices. These specifications are stored in a repository. Device-specific profiles, which represent the configuration specifications, are generated, and the deployment agents of the managed devices configure the device as specified.

CM tools are not the only option to automate or modularize configuration of system landscapes. This may also be achieved by writing custom scripts that automate configuration tasks (Magherusan-Stanciu et al., 2011). However, one advantage is the common language for configuration specifications provided by these tools, which make the specifications sharable for common configuration tasks, even across organizations (cf. (Meyer et al., 2013)).

Managing large system landscapes can also be achieved by configuring virtual machines as templates that are reused. However, centrally updating these machines is not explicitly covered

by this approach. Here, additional concepts and tools would be needed. A similar concept to that of preconfigured virtual machines is that of containers such as Docker (docker.com). Docker containers can run single processes, may be individually spawned and terminated, and may be recombined to form larger applications (Wettinger et al., 2014b).

Configuration management tools also play an important role in agile development projects that use the concept of continuous delivery. Continuous delivery is a practice in which, through automated tests, continuous integration of new features into the main code branch, and continuous delivery of these new features into the production environment depends on tool support (Humble and Farley, 2010). Configuration management tools, along with desktop virtualization technology or infrastructure as a service (IaaS) offerings, are used to approximate a production landscape, without the production landscape's performance requirements, for each developer (cf. (Spinellis, 2012)). This enables the developer to test new features against the main code branch of a software product. The closely related trend of teaming up developers and operations professionals (DevOps) in projects is adequate for projects like those of software as a service (SaaS) (like Google), shrink-wrapped devices (like the iPhone), or customized applications (like SAP ERP) (Spinellis, 2012).

2.2 Related Work

CM tools are also not the only means necessary for managing systems that offer services to their users. CM is also strongly related to the processes of change, release, and deployment management (Rance, 2011, p. 115). As CM tools are used for automating the installation and configuration of application components on nodes, configuration management and deployment management are often used synonymously (Wettinger et al., 2014a). Arcangeli et al. (2015) review work on the automatic deployment of distributed software systems. However, their focus is more on software engineering as they review technologies such as OSGi, which defines a dynamic module system for the programming language Java (Hall et al., 2011). Whereas CM tools are often concerned with infrastructure and platform management aspects, the deployment management tools reviewed by Arcangeli et al. focus on application management aspects.

Rahman et al. (2011) present a taxonomy of components that are required for managing grid computing applications and probe nine grid computing projects against these requirements. The authors conclude that grid workflow management systems are lacking cooperative application scheduling. The systems usually have capabilities for optimization as well as fault discovery, diagnosis, and recovery. According to the authors, configuration and protection functions that cope with the complexity and volatility in a gird computing environment are also needed.

No previous survey of literature on CM tools could be identified, which is why this research is conducted. Its design is presented in the next section.

3. Research Design

In order to answer the research questions, a structured approach is required. The research methodology, aligned with the literature review guidelines by Webster and Watson (2002), is illustrated in figure 1.

3.1 Source and Search

In order to identify literature, twelve internet accessible literature databases were queried with a Boolean search string. The search string requires an article to contain one CM tool name and the words "configuration management". The names of the eleven CM tools originate from the study by Delaet et al. (Delaet et al., 2010) who selected these tools and compared them with their framework.

Relevant publications are expected to name tools, for instance in their related work sections. We acknowledge that not all existing configuration management tools are included in the search string, but works should mention those popular ones named in the survey of Delaet et al. (2010) at least in the related work or research background sections. Therefore, the whole article had to be searched and a full text search was executed in all cases. This approach was selected in favor of searching for "configuration management tools" as authors might use different formulations (e.g. "tools for configuration management", "configuration management systems" etc.) as well as to ensure that the identified papers were not merely discussing CM on a theoretical level without mentioning specific implementations.

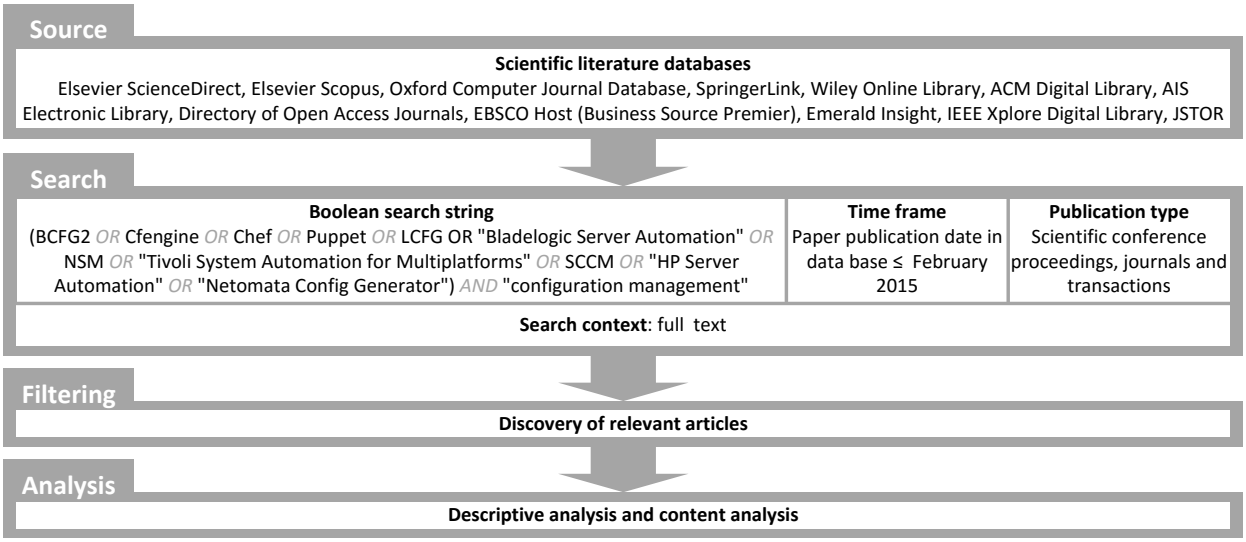

Figure 1: Review process, aligned with Webster and Watson (Webster and Watson, 2002)

No age limit was used for the search. The queries were performed in March 2015. If possible, scientific conference proceedings, journals, or transactions were selected as publication types. Books were not selected, as they usually are not peer-reviewed. Furthermore, only publications in English were considered.

3.2 Filtering and Review

503 publications were retrieved from the databases as displayed in table 1.

These results were then cleaned by deleting the duplicates (121), and publications that were not papers from scientific conference proceedings, journals, or transactions (e.g. proceeding introductions, workshop descriptions, technical reports etc.) (45). Papers that had duplicates were

counted in the database operated by the original publisher. In some cases, papers were found by one database, but not by the one of the publisher, although the search parameters were identical. In these cases, the paper was assigned to the finding database.

Database	initial results	cleaned	relevant
ACM Digital Library	188	121	59
AIS Electronic Library	2	2	1
EBSCO Host (Business Source Premier)	9	4	1
Elsevier ScienceDirect	21	21	10
Elsevier Scopus	151	75	33
IEEE Xplore Digital Library	108	91	49
SpringerLink	15	14	4
Wiley Online Library	9	9	2
Sum	503	337	159

Table 1: Search results per database

After the cleaning step, the papers were manually analyzed by two researchers as full text and irrelevant papers were sorted out. Papers with no focus on CM tools were excluded.
The literature collection process yielded a final number of 159 relevant papers. A file in the BiBteX format with their bibliographic data may be downloaded (http://www.mrcc.ovgu.de/fileadmin/media/documents/config-mgmt-tools.bib).

3.3 Analysis

A descriptive analysis and a content analysis were performed in order to answer the research questions. For the descriptive analysis, the bibliographical data was retrieved and consolidated using the tool JabRef (jabref.sourceforge.net).

During the filtering, the researchers gained a first overview of the papers. This insight was used to inductively create categories to structure the papers. For the content analysis, the papers were analyzed in detail, focusing on their usage of CM tools, notes were taken to summarize their content, and finally they were categorized.

3.4 Limitations

A limitation of the review is that no forward and backward search was performed to identify further relevant literature as Webster and Watson (2002) have suggested. With a backward search, the referenced publications of a paper are also considered to be included into the literature review itself. With the forward search, publications that reference the identified paper are included. Due to the large number of identified relevant publications, this step was not performed for this paper. Furthermore, no manual scanning of tables of contents of journals or conference proceedings was performed.

4. Review

This section presents the findings of the review. In the first subsection the descriptive research questions are answered, and in the second subsection the content-oriented research question is addressed.

4.1 Descriptive Analysis

The first identified publications appear in 1998 as is illustrated in figure 2. However, based on the search scheme, it cannot be stated that these are the first publications. A backward search is necessary. The four publications from 1998 (Lockard and Larke, 1998, Traugott and Huddleston, 1998, Da Silva et al., 1998, Da Silveira and Da Silva, 1998) were published in the proceedings of the twelfth Large Installation System Administration Conference (LISA). Three of them mention the tool LCFG, and one mentions Cfengine (Traugott and Huddleston, 1998). All four papers reference Anderson's presentation of LCFG from 1994 (Anderson, 1994) at the LISA conference of that year. Cfengine's creator Burgess presents Cfengine in 1995 in a journal article of the same organization (USENIX) that organizes the LISA conferences. Therefore, not surprisingly, it can be stated that this research has its origins in the systems administrators' community, and publications on CM tools started in 1994.

There was a first peak in publications in 2005 and then the highest peak in 2013, but other than that, an overall growth of publications can be observed. The drop in 2015 can be explained by the fact that the database query was performed in March 2015. The peak of 2013 can be explained with the trend of cloud computing that exhibited a steady increase of publication starting in 2008 (Yang and Tate, 2012).

Of the 159 publications, 36 were journal articles. No journal has publication rates that clearly differentiate it from the others. Among the conferences the Systems Administration Conference (LISA) has most papers with 33 followed by the IFIP/IEEE International Symposium on Integrated Network Management (IM) with eight publications.

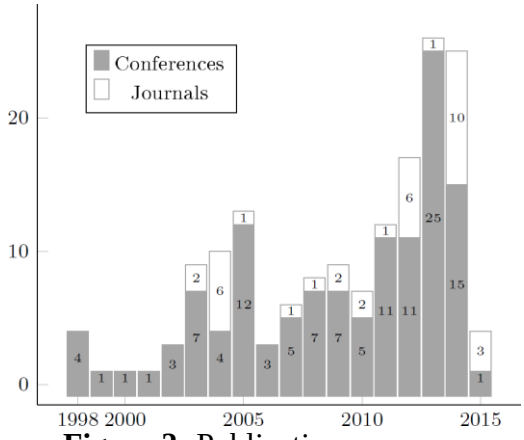

Figure 2: Publications per year

Configuration management tool	Named	In focus
Cfengine	89	15
Puppet	82	11
LCFG	50	7
Chef	46	9
BCFG2	17	3
SCCM	4	1
NSM	2	0
Bladelogic Server Automation	1	0
Tivoli System Automation for Multiplatforms	1	0
HP Server Automation	1	0
Netomata Config Generator	1	0
Self-developed tool	n/a	38

Table 2: Tools in identified papers

378 authors have published papers as authors or co-authors, 63 have published two, 17 have published three, and seven authors have published four papers. Five or more papers either as authors or as co-authors were published by Paul Anderson (12), Wouter Joosen (9, only as co-author), Mark Burgess (7), Frank Leymann (7, only as co-author), Bart Vanbrabant (7), Johannes Wettinger (7), Brad Bradshaw (5, only as co-author), and Thomas Delaet (5).

For all 159 publications it was also recorded which of the eleven CM tools they mentioned. Table 2 shows by how many papers each tool was named and how many papers explicitly focused on a specific tool. This statistic shows that Cfengine, Puppet, and LCFG are the most popular CM tools among researchers. Although Chef is two times more frequently in focus than LCFG, Spinellis (2012) named Chef instead of LCFG as one of the three popular CM tools. Puppet and Chef were only first released in 2005 and 2009, respectively (Wikipedia, 2015). 39 papers reported on tools that were developed by the authors and are not among the eleven tools that were used for searching publications.

The next section presents the content of the papers.

4.2 Content Analysis

The categories for the content analysis were created inductively as described in the research design section 3. They are displayed and quantified in figure 3. In figures 4 and 5 they are presented with by means of sample papers.

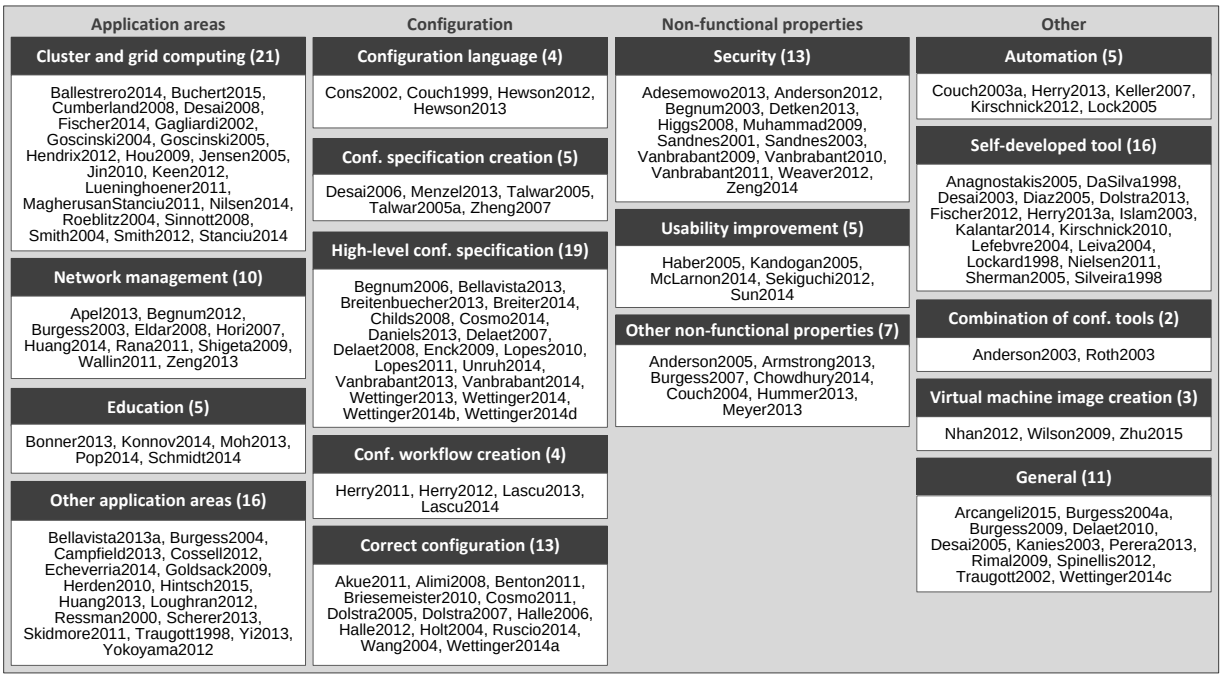

Figure 3: Papers sorted into the categories

(papers are identified by the BibTeX IDs, in parentheses the number of papers for each category is denoted. BibTeX file may be downloaded here: http://www.mrcc.ovgu.de/fileadmin/media/documents/config-mgmt-tools.bib)

Application areas: Several papers describe applications of CM tools in different areas. Reoccurring areas were those of cluster and grid computing, network management, educational settings, as well as several other application areas.

Configuration: Another group of papers present research on configuration aspects. One category is language of configuration specifications. Configuration specifications can be very complex and therefore the creation of configuration specifications is another category. One reason for the complexity is that most configuration specifications are low-level. Consequently, research on high-level configuration specification aims at describing the configurations on a level that is closer to the actual business requirements of a system landscape. Configuration workflow creation is addressed by a small group of papers. Finally, ensuring that a correct configuration is used is also researched.

Non-Functional Properties: Several non-functional properties are researched: security concerns, CM tool usability improvements, and other non-functional properties.

Other: Five further categories were created that could not be grouped with the others: First, an automation category that includes papers discussing automation aspects for CM tools. Second, work on self-developed CM tools that are not among the eleven tools used for the paper search. Third, a category with papers that combine configuration tools is formed. Fourth, CM tools are also used in virtual machine image creation, and fifth, a general category of papers was used to categorize papers that do not fit into other categories.

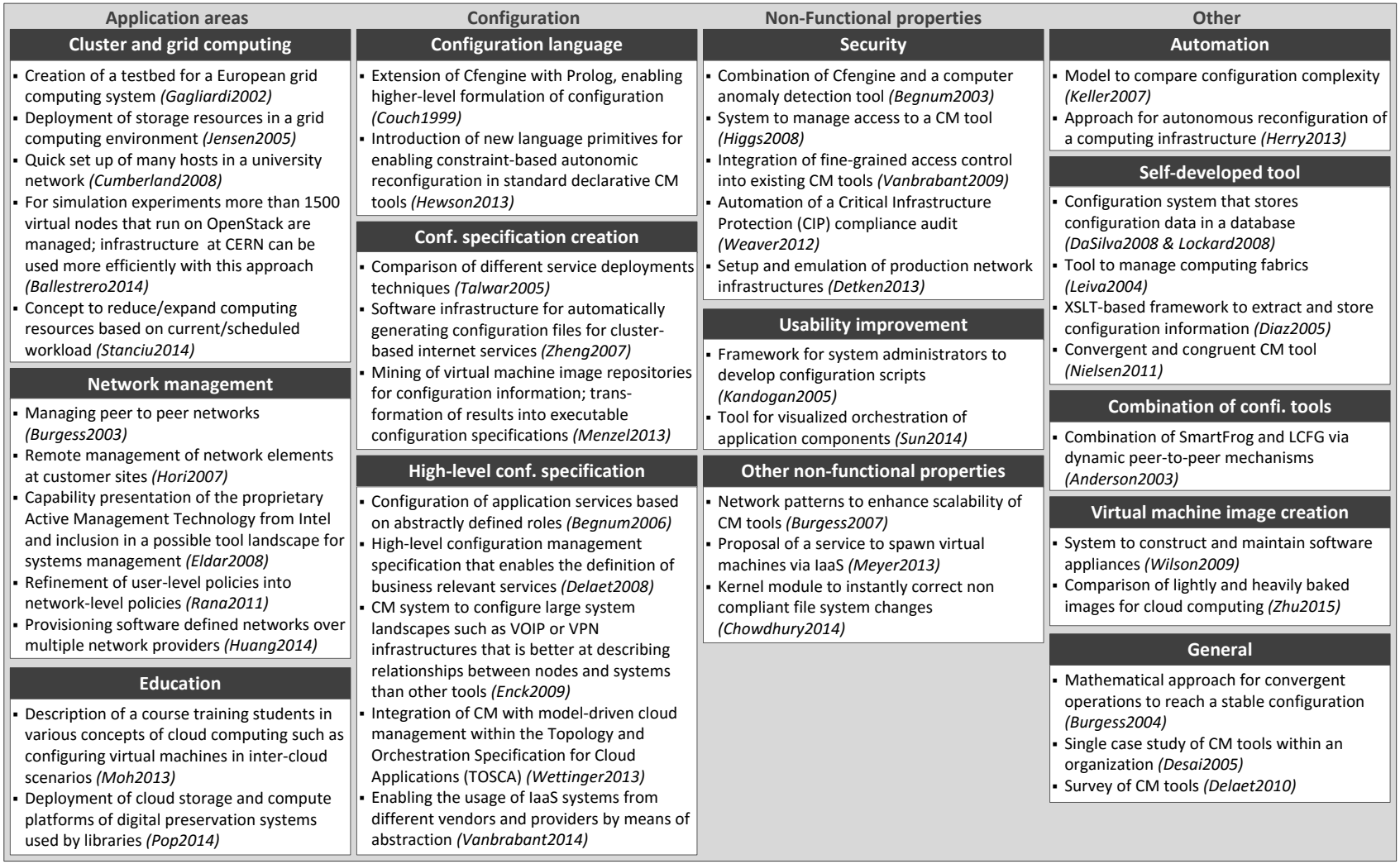

Figure 4: Categories presented by means of sample papers

(papers are identified by the BibTeX IDs.

BibTeX file may be downloaded here: http://www.mrcc.ovgu.de/fileadmin/media/documents/config-mgmt-tools.bib)

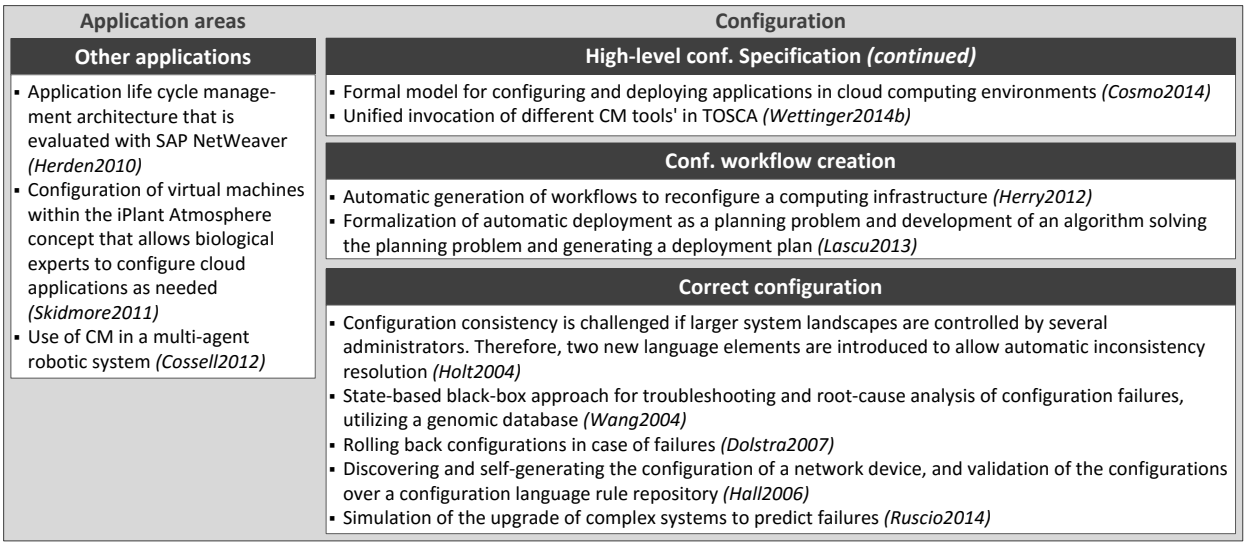

Figure 5: Categories presented by means of sample papers *(continued)*

The publications may address different categories. However, publications were only categorized by those categories that were its main focus, according to our assessment after analyzing the publications in detail.

5. Discussion

The first peer-reviewed publication on configuration management tools appeared in 1994. It was Anderson's presentation of the CM tool LCFG. His motivation for designing LCFG was to configure physical machines that had a default vendor's configuration in accordance to the requirements of the organization using the machines, in his case Edinburgh University's computer science department.

A first peak of publications could then be seen in 2004 and 2005, where several publications on cluster and grid computing appeared, which is also the most researched topic of the reviewed publications. Focus now was not anymore only on lab computers and individual servers, but on nodes working together in a cluster or grid.

The second peak of publications was between 2011 and 2014. This followed the advent of cloud computing in 2008, a supposedly more business oriented computing paradigm (Misra and Mondal, 2011). Cloud providers offer application services to customers and configuration management tools are used to configure the underlying infrastructure, but also the application services themselves. How to move applications from one cloud platform to another and how to define configuration on a higher-level is researched by several authors as demonstrated by the fact that most publications during this second peak focused on high-level configuration specification.

Anderson's 1994 publication was presented at the Large Installation System Conference (LISA) and system administration has been a large driver of CM tool development. But, what has not been researched much is the role of CM tools in organization's information system (IT systems,

processes, and people). Only Desai et al. (2005) have looked at how to implement configuration management in an organizational setting. We would have expected to see more behavioral or interdisciplinary studies because, as illustrated in the introduction, configuration management tools can be a source for other management domains such as risk management. That information systems researchers have not embraced this topic is also indicated by only one relevant work found in the AIS Electronic Library.

Cfengine, Puppet and LCFG were the most frequently named tools in the surveyed works. Nonetheless, stating that instead of LCFG Chef should be named as one of the three most popular tools (Spinellis, 2012) seems justified, considering that Chef had been released more than a decade after LCFG and was nearly as many times mentioned as LCFG. But, these tools seem not to be the answer for everyone as self-developed tools were still being presented when these tools had long been around. For instance, Kalantar et al. in 2014 (2014) present the tool weaver that can describe the desired state of an environment going beyond the popular tools' initial scope of configuring individual nodes. High-level configuration specification, where node relationships need to be addressed in order to provision complete services that may span several nodes, has been one of the most researched topics and appears to still need research as some of the most recent publications address this topic.

Delaet et al. (2010), having surveyed the eleven CM tools, presented their vision of a CM tool being able to manage the complete range of devices from desktop computers, over servers to laptops and smart-phones. Several works have addressed network management, including management of network devices, but no work could be identified that also focused on managing smart phones. The dominating smart phone operating systems Android and iOS have deployment and configuration mechanisms that are fundamentally different from those of Unix derivatives or Windows systems. Potentially, the announcement by Microsoft to have apps that can be run on all Windows based device types may lead in that direction. However, published research on commercial, closed source products has been sparse.

The significance of CM tools is further underlined by several works that presented the introduction of CM tools into the curricula of computer science students.
Some of the points discussed in this section make paths of future work promising. These will be outlined in the next and last section.

6. Conclusion

High-level configuration specification stands out as a topic studied by researchers in the area of configuration management tools. A vision is to be able to have a non-technical configuration management, where services can be orchestrated together with the ease of a mouse click. In practice, this vision is addressed from different perspectives: puppetlabs announced orchestration functionality (puppetlabs.com/puppet/puppet-application-orchestration-news) for their tool. But, for this purpose also specific PaaS offerings exist. On jujucharms.com customers may click together services and deploy them to any private or public cloud. Furthermore, the IaaS platform OpenStack, which is described as a de-facto standard (Forrester, bit.ly/1k6UCts), offers its own orchestration functionality with its heat project (wiki.openstack.org/wiki/Heat), and it integrates with Puppet and Chef. So, from a vendor perspective, high-level configuration or orchestration is addressed differently.

OASIS with TOSCA attempts to standardize orchestration. The reviewed works around TOSCA introduce an additional layer around tools such as Puppet and Chef in order to invoke them from a TOSCA-compliant execution engine (Wettinger et al., 2014a). This is pragmatic as standardizing the configuration languages of Puppet and Chef does not lie in the power of those authors or that of OASIS. But, a standard configuration language without additional complexity layers would be preferable. Moving in that direction are large companies such as Red Hat or VMware with their Open Container Project (opencontainers.org) that shall specify industry standards for container formats and their execution engines. Towards a better comparison of the different approaches to orchestration, a framework should be created in future work, giving advice to practitioners which technology to choose from.

As described above, published research on commercial, closed source products is sparse. However, configuration of enterprise systems such as commercial ERP products is a major challenge for configuration management, but has not been addressed in the reviewed publications. This gap should also be addressed in future work.

The CM tool landscape is becoming more comprehensive with the availability of advanced testing means thereby better supporting workflows common in mature engineering disciplines. However, there is still progress to be made to cover the full life-cycle. For instance, it would be beneficial to have integrated human-readable modeling means for cloud application services in a design stage. Such modeled cloud application services are then configured in the implementation stage and so on.

Methodologically, this review should be extended by including a back- and forward search. This may yield a higher coverage of relevant papers as different nomenclature may have been used by papers that did not match the search string, and thus were not found. Based on the identified predominant CM tools, a selective investigation of further developments within the area of CM tools might be useful since practitioners' discussions in mailing lists, blogs, as well as general Q&A websites may contain valuable state of the art information.

References

Anderson, P. (1994) "Towards a High-Level Machine Configuration System" in 8th Conference on System Administration, LISA, Berkeley, CA, USA: USENIX, pp. 19–26.

Arcangeli, J.-P., Boujbel, R., and Leriche, S. (2015) "Automatic Deployment of Distributed Software Systems: Definitions and State of the Art", Journal of Systems and Software, (103)0, pp. 198–218.

Da Silva, F. Q. B., Da Cunha, J. S., Franklin, D. M., Varejão, L. S., and Belian, R. (1998) "An NFS Configuration Management System and Its Underlying Object-Oriented Model" in 12th Conference on System Administration, LISA, Berkeley, CA, USA: USENIX, pp. 121–130.

Da Silveira, G. E. and Da Silva, F. Q. B. (1998) "A Configuration Distribution System for Heterogeneous Networks" in 12th Conference on System Administration, LISA, Berkeley, CA, USA: USENIX, pp. 109–120.

Delaet, T., Joosen, W., and Van Brabant, B. (2010) "A Survey of System Configuration Tools" in van Drunen, R. (eds.) 24th Large Installation System Administration conference, LISA, USENIX, pp. 1-14.

Desai, N., Bradshaw, R., Matott, S., Bittner, S., Coghlan, S., Lueninghoener, C., Leggett, T., Navarro, J.-p., Rackow, G., Stacey, C., and Stacey, T. (2005) “A Case Study in Configuration Management Tool Deployment” in Blank-Edelman, D. N. (eds.) 19th Conference on Large Installation System Administration, LISA, USENIX, pp. 39–46.

Fischer, J., Knepper, R., Standish, M., Stewart, C. A., Alvord, R., Lifka, D., Hallock, B., and Hazlewood, V. (2014) “Methods For Creating XSEDE Compatible Clusters” in 2014 Annual Conference on Extreme Science and Engineering Discovery Environment, XSEDE, ACM, pp. 74:1–74:5.

Hall, R., Pauls, K., McCulloch, S., and Savage, D. (2011) OSGi in Action: Creating Modular Applications in Java, Stamford, CT, USA: Manning.

Humble, J. and Farley, D. (2010) Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation, Boston, USA: Pearson Education.

Kalantar, M. H., Rosenberg, F., Doran, J., Eilam, T., Elder, M. D., Oliveira, F., Snible, E. C., and Roth, T. (2014) “Weaver: Language and runtime for software defined environments”, IBM J. Res. Dev., (58)2-3, pp. 10:1–10:12.

Lockard, J. and Larke, J. (1998) “Synctree for Single Point Installation, Upgrades, and OS Patches” in 12th Conference on System Administration, LISA, Berkeley, CA, USA: USENIX, pp. 261–270.

Magherusan-Stanciu, C., Sebestyen-Pal, A., Cebuc, E., Sebestyen-Pal, G., and Dadarlat, V. (2011) “Grid System Installation, Management and Monitoring Application” in 10th International Symposium on Parallel and Distributed Computing, ISPDC, IEEE, pp. 25–32.

Meyer, S., Healy, P., Lynn, T., and Morrison, J. (2013) “Quality Assurance for Open Source Software Configuration Management” in 15th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC, IEEE, pp. 454–461.

Misra, S. C. and Mondal, A. (2011) “Identification of a company’s suitability for the adoption of cloud computing and modelling its corresponding return on investment”, Mathematical and Computer Modelling, (53)3–4, pp. 504–521.

Rahman, M., Ranjan, R., Buyya, R., and Benatallah, B. (2011) “A Taxonomy and Survey on Autonomic Management of Applications in Grid Computing Environments”, Concurrency and computation: practice and experience, (23)16, pp.1990–2019.

Rance, S. (2011) ITIL – Service Transition, Norwich, UK: The Stationery Office.

Schaefer, A., Reichenbach, M., and Fey, D. (2013) “Continuous Integration and Automation for Devops” in Kim, H. K., Ao, S.-I., and Rieger, B. B. (eds.) IAENG Transactions on Engineering Technologies, volume 170 of Lecture Notes in Electrical Engineering, Springer, pp. 345–358.

Spinellis, D. (2012) “Don’t Install Software by Hand”, IEEE Software, (29)4, pp.86–87.

Traugott, S. and Huddleston, J. (1998) “Bootstrapping an Infrastructure” in 12th Conference on System Administration, LISA, Berkeley, CA, USA: USENIX, pp. 181–196.

Webster, J. and Watson, R. T. (2002) “Analyzing the Past to Prepare for the Future: Writing a Literature Review”, MIS Quarterly, (26)2, pp. xiii–xxiii.

Wettinger, J., Behrendt, M., Binz, T., Breitenbücher, U., Breiter, G., Leymann, F., Moser, S., Schwertle, I., and Spatzier, T. (2013) “Integrating Configuration Management with Model-driven Cloud Management based on TOSCA” in 3rd International Conference on Cloud Computing and Service Science, CLOSER, SciTePress, pp. 437–446.

Wettinger, J., Binz, T., Breitenbücher, U., Kopp, O., and Leymann, F. (2014a) “Streamlining Cloud Management Automation by Unifying the Invocation of Scripts and Services Based

on TOSCA", International Journal of Organizational and Collective Intelligence, (4)2, pp. 45–63.
Wettinger, J., Breitenbücher, U., and Leymann, F. (2014b) "Compensation-Based vs. Convergent Deployment Automation for Services Operated in the Cloud" in Franch, X., Ghose, A., Lewis, G., and Bhiri, S. (eds.) Service-Oriented Computing, volume 8831 of Lecture Notes in Computer Science, Springer, pp. 336–350.
Wikipedia (2015) "Comparison of Open-Source Configuration Management Software", accessed: 14.07.2015.
Yang, H. and Tate, M. (2012) "A Descriptive Literature Review and Classification of Cloud Computing Research", Communications of the Association for Information Systems, (31), pp. 35–60.