An Interactive Interface for Bulk Software Deployment in IoT

Farshid Tavakolizadeh
Fraunhofer Institute for Applied Information Technology
Sankt Augustin, Germany
farshid.tavakolizadeh@fit.fraunhofer.de

Hanbing Zhang
RWTH Aachen University
Aachen, Germany
hanbing.zhang@rwth-aachen.de

Sisay Adugna Chala
Fraunhofer Institute for Applied Information Technology
sisay.adugna.chala@fit.fraunhofer.de

ABSTRACT

Bulk software deployment is a tedious and error-prone task. This has prompted concerns by the advent of the Internet of Things (IoT) into daily lives requiring recurrent deployment of software to a large number of heterogeneous devices. This work proposes an interactive graphical user interface to simplify common software deployment activities in IoT systems. The results of laboratory usability testing show high system usefulness and overall user satisfaction.

CCS CONCEPTS

• **Software and its engineering** → *Software configuration management and version control systems*; *Software maintenance tools*; • **Human-centered computing** → *User centered design.*

IoT2019, October 22–25, 2019, Bilbao, Spain

ACM ISBN 978-1-4503-7207-7/19/10.
https://doi.org/10.1145/3365871.3365912

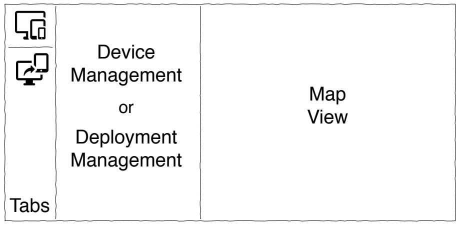
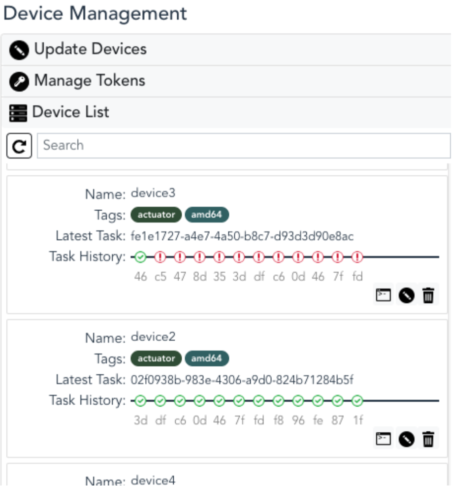


Figure 1: User interface layout, composed of three column panels.



Name: device3
Tags: actuator amd64
Latest Task: fe1e1727-a4e7-4a50-b8c7-d93d3d90e8ac
Task History: 46 c5 47 8d 35 3d df c6 0d 46 7f fd

Name: device2
Tags: actuator amd64
Latest Task: 02f0938b-983e-4306-a9d0-824b71284b5f
Task History: 3d df c6 0d 46 7f fd f8 96 fe 87 1f

Name: device4

Figure 2: Device Management panel.

ACM Reference Format:
Farshid Tavakolizadeh, Hanbing Zhang, and Sisay Adugna Chala. 2019. An Interactive Interface for Bulk Software Deployment in IoT. In *9th International Conference on the Internet of Things (IoT 2019), October 22–25, 2019, Bilbao, Spain.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3365871.3365912

INTRODUCTION

The advancements of technology in recent years have led to lower manufacturing costs and paved the path to pervasive connected devices. The number of connected devices is expected to reach 75.4 billion by 2025 [2]. The estimates have turned into a global issue for software and security experts concerned with technology readiness for such a scale [6, 9]. Some [9] consider the extremely dynamic nature of Internet of Things (IoT) systems as a cause for many new challenges for software deployment, testing, and debugging.

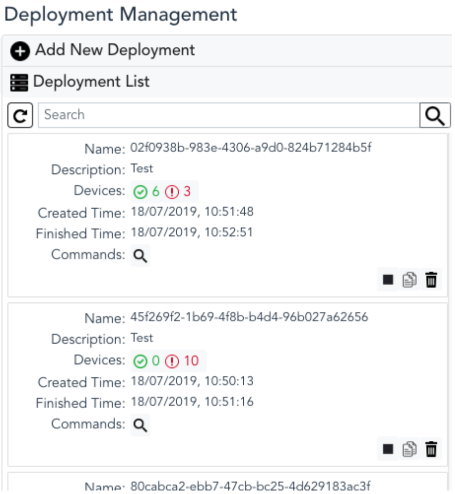
Software deployment activities involving similar steps often depend upon customizations on heterogeneous devices. On top of that, a simple deployment which requires minimal efforts can get very tedious when repeated on several devices of the same type. As such, people involved in software deployment often rely on tools to reduce the complexity and increase operational efficiency.

This work tries to minimize the time developers spend doing mundane tasks of deployment. It proposes a graphical user interface (GUI) addressing the whole deployment pipeline as suggested by CPSwarm [1] to i) efficiently select designated devices in large scale environments, and ii) roll-out deployments and intuitively monitor the status of the system during and after the deployment. The work inclines towards two trending IoT segments with high popularity: smart cities and connected buildings [8].

RELATED WORK

Software deployment is a widespread process involved in the development and maintenance of most software systems. Mender [7] and Eclipse hawkBit [3] are deployment management tools specifically developed for IoT systems. Both provide GUIs capable of maintaining and deploying software as pre-built packages [7] or by the selection of files [3]. Target addressing on these systems is based on IP addresses, domain names, aliases, or tags. Moreover, Mender recognized the need to visualize deployment status in an intuitive way and displays device status icons in a compact, two-dimensional form. It also shows a summary in clusters of successes and failures. Similarly, the state-of-the-art software configuration management works [4, 11] alleviate usability issues using GUIs. Their interfaces enable users to deploy packages on targets using IP addresses, names, or tags. Besides, they present the status of deployments and visually convey the total number of failures.

Existing works offer practical solutions useful for the deployment of software in IoT environments. Nonetheless, they fail to address important usability aspects of an interface for deployment on large-scale IoT systems. In particular, none of the reviewed tools treat devices as objects with geo-positional characteristics that may be the only way to target them. Moreover, they offer minimal monitoring capabilities which quickly become counter-intuitive when the number and diversity of failures increase.



Deployment Management

Add New Deployment

Deployment List

Search

Name: 02f0938b-983e-4306-a9d0-824b71284b5f
Description: Test
Devices: ✓ 6 ! 3
Created Time: 18/07/2019, 10:51:48
Finished Time: 18/07/2019, 10:52:51
Commands:

Name: 45f269f2-1b69-4f8b-b4d4-96b027a62656
Description: Test
Devices: ✓ 0 ! 10
Created Time: 18/07/2019, 10:50:13
Finished Time: 18/07/2019, 10:51:16
Commands:

Name: 80cabca2-ebb7-47cb-bc25-4d629183ac3f

Figure 3: Deployment Management panel.

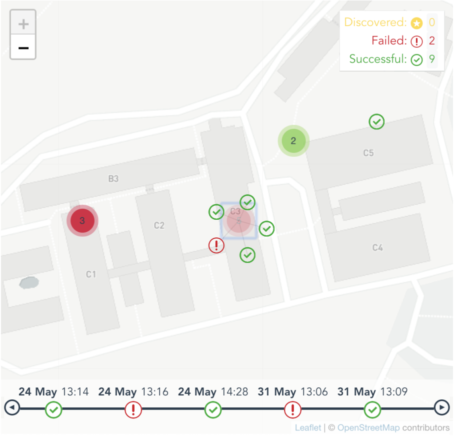


Figure 4: Map View panel showing 11 devices, 10 of which are clustered at this zoom level.

DESIGN AND IMPLEMENTATION

This work proposes a graphical interface developed in three phases. The first phase built the initial idea and presented high-level concepts in the form of a low-fidelity prototype. Five domain experts participated in formative evaluations to validate the usefulness of the concepts. The user feedback was applied successively into a medium-fidelity prototype. The same group of users evaluated this prototype and provided feedback to early refinements. The final phase resulted in a high-fidelity prototype presented here. The source code of the presented work along with a backend component that supports the realization of deployment operations on real devices are available as open source [10].

Figure 1 shows the proposed layout of the GUI. It is composed of three columns: Tabs for switching between functionalities, Device Management/Deployment Management, and the Map View.

Device Management (Figure 2) includes utilities mostly related to device registration. Devices are added through discovery when connected for the first time, carrying a valid token issued previously via the UI. Users can modify the meta-information (e.g. tags, location) or delete a device to revoke its access. Device location is set manually by the user or produced by external positioning sources. Devices with location are displayed on the **Map View** (Figure 4), clustered in case of proximity. This is particularly practical during deployment and monitoring. For hands-on debugging, users can open a terminal and gain Shell-like access to each device. Furthermore, each device has a timeline showing the history and status of past deployments.

Deployment Management (Figure 3) consists of features related to package build, transfer, installation, and execution. For the build, the user may select source files, provide build commands, and select a device or an external host (e.g. a docker container with build resources) to perform the build. This is to produce artifacts that are stored for later reuse and may be transferred to selected targets. Device names and tags are used to select target devices individually or in bulks. It is also possible to utilize the **Map View** (Figure 4) to pick a device or a cluster of them. Furthermore, the user may supply installation and execution commands which are used to prepare the package and execute included applications. The aforementioned steps are only the configurations of the deployment. Once the user triggers the deployment, an intuitive progress tree shows the status of the deployment process; see Figure 5. The levels correspond to deployment stages and nodes represent devices. Clicking on a device-node opens the available logs for the deployment on that device. Failed deployments cluster separately based on the similarity of error logs. The status of the deployments is also visible on the **Map View** (Figure 4) giving location-based insights into current deployments.

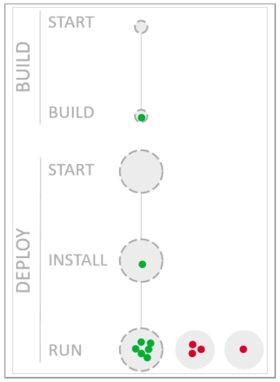


Figure 5: Deployment progress tree for one build host and 11 devices. The build is successful, installation is in progress (1) or finished (10). Four devices are in a failed run-time state, three of them with similar errors.

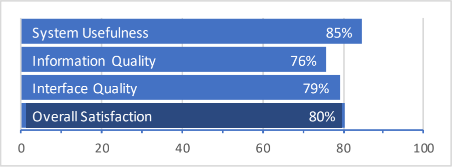


Figure 6: Usability scores and overall satisfaction for the final prototype.

ACKNOWLEDGMENTS

This work is supported by the European Commission through the CPSwarm H2020 project (https://cpswarm.eu) under grant no. 731946.

EVALUATION & DISCUSSION

We prepared a motion-triggered light system to test the proposed UI in practical settings. The setup composed of three Raspberry Pi devices, one connected to a passive infrared (PIR) motion detector, and two connected to LED lights. We also prepared two C++ programs, one for reading motion data and sending to a broker and another for getting the data and triggering the lights. We randomly invited 10 participants from a team of 25 software engineers with software deployment experience for connected buildings and smart cities. The participants were tasked to use the interface to setup the build, installation, and execution of the programs on the selected targets. Overall, they engaged with most offered UI features to diagnose and resolve unexpected deployment issues.

After successfully completing the deployments, we asked them to fill up a Post-Study System Usability Questionnaire (PSSUQ) [5] to validate the system usefulness, information, and interface quality with regards to the given tasks. Figure 6 shows the results suggesting high system usability in the laboratory setting. This indicates that the proposed UI elements have a high potential for use in deployment and monitoring interfaces within IoT systems. Nevertheless, it demands further work to assess the information quality at scale and prove the usability under real-life conditions.

REFERENCES

[1] Alessandra Bagnato, Regina Krisztina Bíró, Dario Bonino, Claudio Pastrone, Wilfried Elmenreich, René Reiners, Melanie Schranz, and Edin Arnautovic. 2017. Designing swarms of cyber-physical systems: the H2020 CPSwarm project. In *Proceedings of the Computing Frontiers Conference*. ACM, 305–312.

[2] Louis Columbus. 2016. Roundup of internet of things forecasts and market estimates, 2016. *Forbes Magazine, New York* (2016).

[3] Eclipse Foundation. [n.d.]. Eclipse hawkBit - Update Server. https://www.eclipse.org/hawkbit Accessed: 2019-05-18.

[4] Lorin Hochstein and Rene Moser. 2017. *Ansible: Up and Running: Automating Configuration Management and Deployment the Easy Way*. "O'Reilly Media, Inc.".

[5] James R Lewis. 1992. Psychometric evaluation of the post-study system usability questionnaire: The PSSUQ. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 36. 1259–1260.

[6] Rwan Mahmoud, Tasneem Yousuf, Fadi Aloul, and Imran Zualkernan. 2015. Internet of things (IoT) security: Current status, challenges and prospective measures. In *ICITST-2015*. IEEE, 336–341.

[7] Mender. [n.d.]. Open source OTA software and firmware updates for embedded Linux. https://mender.io/ Accessed: 2018-11-03.

[8] Padraig Scully. 2018. The Top 10 IoT Segments in 2018-Based on 1,600 Real IoT Projects. *IoT Analytics: Market Insights for the Internet of Things* (2018).

[9] Antero Taivalsaari and Tommi Mikkonen. 2017. A roadmap to the programmable world: software challenges in the IoT era. *IEEE Software* 34, 1 (2017), 72–80.

[10] Farshid Tavakolizadeh and Hanbing Zhang. 2019. CPSwarm Deployment Tool. https://github.com/cpswarm/deployment-tool. Accessed: 2019-05-30.

[11] Sricharan Vadapalli. 2018. DevOps: Continuous Delivery, Integration, and Deployment with DevOps: Dive into the core DevOps strategies. Packt Publishing, 101–103.