A deployment approach for Smart Building applications

Imen Abdennadher, Ismael Bouassida Rodriguez and Olfa Awled Al Hadj Abdalah

ReDCAD, University of Sfax B.P. 1173, 3038, Sfax, Tunisia

Abstract

Smart Buildings have been the subject of research for more than three decades. Since Smart buildings are considered as the next generation of living and working environments, their development and deployment have become a necessity. In ubiquitous computing systems, deploying Smart Building applications is a challenge due to the important number of software components that these applications contain. To execute such applications, their software components must be deployed on different devices in their target environments. This process is called software deployment. In this work, we propose a deployment approach for Smart Building applications. The deployment approach aims at transforming the Smart Building application from an application deployed on a single machine to an application deployed on several machines through the use of a deployment service. This service supports the redeployment at runtime enabling the Smart Building application to adapt itself in order to handle the changes that have been occurring during the application execution.

Keywords

Smart Buildings, software deployment, distributed environment, software components

1. Introduction

In the past, software applications were monolithic applications (application consisting of a single component to deploy on a single machine). In recent years, current software applications have become more and more complex and no more monolithic. They may consist of a huge number of different components distributed over many computers connected through the network and assembled as a system and operating together. The aim of the software deployment process is to place an already developed application into its target environment and make it available for use and then keep it operational. Also, software deployment was managed manually. Users need to install their applications on their computers, using a floppy disk to copy an executable program to another computer. However nowadays, a deployment process needs a level of automation due to the increasing number of devices. Smart Building application is among the software applications. The aim of Smart Buildings is improving the life quality of their occupants. To manage a Smart Building it is necessary to develop a Smart Building application. The deployment of Smart Building applications becomes too complex to be manually managed due to its huge number of components which are distributed over the network .

Tunisian Algerian Conference on Applied Computing (TACC 2021), December 18–20, 2021, Tabarka, Tunisia

imen.abdennadher@redcad.org (I. Abdennadher); bouassida@redcad.org (I. Bouassida Rodriguez); abdalaholfa@gmail.com (O. Awled Al Hadj Abdalah)

In this work, we present a Smart Building case study and its associated application Smart Building Application (SBA). The SBA was first deployed on a single machine, so we apply our deployment approach in order to make the application distributed and to deploy this application on several machines using a deployment service. At runtime, this service supports the redeployment enabling the SBA to adapt itself in order to handle the changes that have been occurring during the application execution. This paper is organized as follows: Section II presents some concepts and definitions related to our work. In section III, we present the state of the art. We describe the Smart Building application case study in section IV. In section V, we present our deployment approach for Smart Buildings. In section VI, we present a deployment scenario of the Smart Building application. Section VII concludes the paper and gives some directions for future work.

2. Concepts and definitions

2.1. Smart Building

Smart Buildings (or intelligent buildings) have been researched and developed over the last three decades [1]. There have been diverse discussions about the Smart Building concept. In the Smart Building dictionary, Paul Ehrlich [2] defines a Smart Building as "a building that integrates technology and process to create a facility that is safe, more comfortable and productive for its occupants, and more operationally efficient for its owners."

Other definitions of the Smart Building concept are as follows:

"An automated system that: Senses what is going on indoors and outdoors. Reacts to ensure, in the most efficient way, safe and comfortable stay in it minimizing amount of energy and resources consumed."[3]

" An intelligent building is a computer aided (automated) building that is designed and centrally managed to ensure safety, comfort and productivity for its occupants as well as energy efficiency, through sensing and communication devices."[4]

"Address both intelligence and sustainability issues by utilizing computer and intelligent technologies to achieve the optimal combinations of overall comfort level and energy consumption."[5]

Consequently, it is evident that a Smart Building has a lot of definitions, it is understood differently by different people, but all these various definitions share some common points that a Smart Building is :

- an automated or computerized building,
- centrally controlled and managed,
- designed to ensure safety, comfort and productivity for its occupants,
- able to sense and react to the environment change,
- aims to ensure resources usage.

2.2. Software deployment

Software deployment is a crucial step in the software life cycle. It happens between the production and execution of software. It is a complex process that covers all post-development activities required to place an application into its target environment and to make it available for use [6].

2.3. Deployment process

In the literature, different activities were defined for the deployment process. Arcangeli et al.[7] defined a set of deployment activities, referring to Carzaniga et al[8]. The proposed process activities are presented in Figure 1:

- Release: concerns all the operations needed to prepare the software component(s) for assembly and distribution (assembling into packages). The result of this activity is an archive (package) that contains the software components. This package will be deployed later on one or more deployment target hosts.
- Installation: this activity allows to insert the software component(s) into one or more deployment target sites. It requires the software component(s) to be transferred (delivery) and configured in order to prepare it (them) for activation.
- Activation: this activity covers all the operations required to start the software system or to install triggers that will launch the software system at an appropriate time.
- Deactivation: refers to the activity of shutting down any executing components of the installed system. In general, Deactivation is required for other activities such as update.
- Deinstallation: removes the software component(s) from the consumer site(s).
- Retire: concerns all the operations done on the producer site, by the software producer in order to mark the software as obsolete. The consequence of retiring a software is that no new version will be produced (however, current or previous versions may remain usable).

After installation, some operations are necessary to check the deployed software and provide evolutions.

- Update: a release of a new version of a software component by the producer. It consists in replacing the old component by a new releases.
- Adaptation : is triggered by an environmental change (user needs, resource variability, etc). The software must be adapted in order to remain operational.
- Reorganization: modifies the logical structure of the system of components, by replacing a component or modifying configuration parameters and/or links between components.
- Redistribution: modifies the deployment plan. It may be required when there is a change of the network topology. It possibly demands a new location for the component(s) to be chosen, and consists in moving the component(s) while preserving constraints, and requirements. The term “redeployment" is used as a synonym for redistribution.

2.3.1. Deployment strategies

In the literature, software deployment can be achieved by two types of strategies: push and pull.

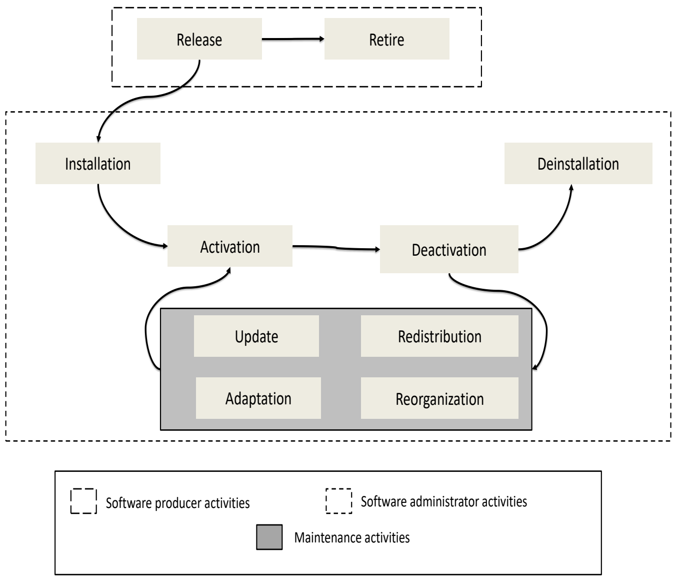

Figure 1: Life cycle of the deployment process [7].

- Push: The deployment process is started by an administrator or a supervisor node. Software components are transferred to deployment nodes without a previous request. In general, the supervisor node contains a program that offers a GUI. An administrator selects software components and deployment nodes. On the other side (the deployment nodes), there is a program that receives transferred services.
- Pull: In opposition to the push strategy, the deployment node starts the deployment process. The user sends a deployment request, and if there is a favorable response, the searched software component will be deployed its node.

2.3.2. Deployment categories

Two deployment categories are identified: static and dynamic deployment, called also respectively: explicit and implicit.

- Static (explicit): The relation between a service or a component and a node is explicitly defined by the administrator. This relation is realized before starting the deployment process.
- Dynamic (implicit): The choice of a service or a component and a node is done in an intelligent and automatic way during the deployment process.

2.3.3. Deployment control

Deployment control can be centralized or decentralized.

- Centralized: A central machine performs the deployment process. Deployment machines are not autonomous and strictly depend on this machine. The centralized architecture is composed of a set of deployment machines and a single deployment manager. The deployment machines are machines on which software components will be installed. The deployment manager manages the communication between the deployment machines. It is also responsible of choosing software components suited to a particular machine.
- Decentralized: There is no central machine to perform the deployment process. Each machine contains a deployment manager which interacts with neighboring machines. This schema is well suited with pervasive computing applications.

2.3.4. Deployment modes

Two deployment modes are distinguished: initial and ulterior.

- Initial: refers to the first deployment of the application in the target environment.
- Ulterior: called also reconfiguration or redeployment. Reconfiguration is done during the system runtime process when changes in the environment appear.

3. Software deployment tools

After presenting the deployment process, we present in the following a set of software deployment platforms. The deployment tools are deployment tools of application on a single host and deployment tools of distributed applications. Moreover, those tools are classified into tools that are independent of the context and other ones which depend on the context.

We have classified the study of this work into two deployment-related work-groups:

- Automatic deployment independent of the context
- Automatic deployment depend on the context

4. Related work

In the software deployment process, several tools are proposed to ensure the deployment of software. There are many programs that are used to install and uninstall software systems from a single machine. Examples include InstallShield [8] and Microsoft Windows [9] Installer. These tools are not more than a compression tool with a user-friendly interface (e.g., a wizard). In these tools, different files of software systems are compressed into self-installing archives and are delivered to users. Users can install, uninstall and configure the software manually. They are intended only for Windows operating systems.

DeployWare [10] is a framework for the deployment of distributed software automatically on large scale infrastructures such as grids . This tool offers a language dedicated to the field of deployment and a virtual machine for this language. This virtual machine is called Fractal Deployment Framework (FDF). The deployment unit is an archive that contains the deployment application and descriptor. The different deployment activities offered by this tool are: the installation, activation, deactivation and uninstallation.

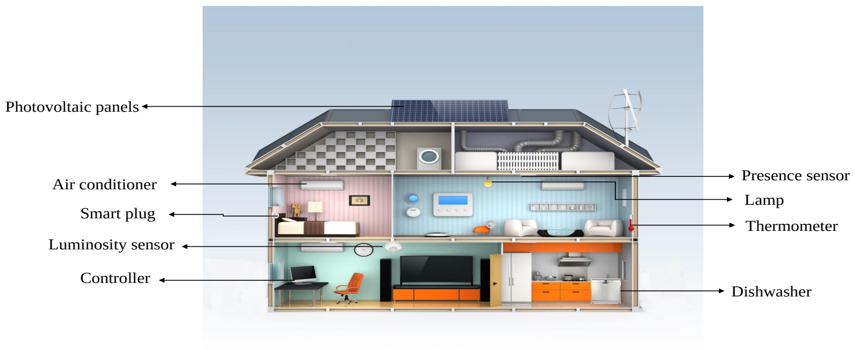

Figure 2: Case study: Smart Building.

SmartFrog [11] is an advanced deployment framework, dedicated to grid infrastructures (composed of distributed and various number of machines). It manages the deployment of distributed systems of components, based on a specific component model. As systems change over time, it is possible to add or remove components at runtime. SmartFrog supports several life cycle activities: installation and uninstallation, activation and deactivation, and reorganization.

The OSGI framework (Open Services Gateway initiative) based on the Java technology [12]. Deployment within OSGi allows to install, deinstall, activate ("start" in OSGi), deactivate ("stop" in OSGi), and update components at runtime without restarting the whole system. It allows the deployment of Java-based components called bundles. A bundle is a Java compressed JAR file that contains a manifest and Java class files, and other resources (libraries, files, etc.).

Kalimucho is a distributed platform for dynamic deployment on heterogeneous devices such as desktops, laptops and mobile devices [13] . The parameters supported by this tool are the kind of devices, and their amount of energy, CPU workload and available memory. Kalimucho supports consumer-side deployment activities: installation and uninstallation, update, reorganization and redistribution.

According to this study of the existing tools that ensure the deployment of the softwares, we opt to use the OSGI framework because it offers very promising functionalities such as it easily allows the deployment of Java applications composed of one or more OSGi bundles on heterogeneous devices, such as personal computers, smartphones, and tablets.

5. Case study: Smart Building

To illustrate the use of our deployment approach, we introduce in the following an example of a Smart Building illustrated in Fig. 2. The Smart Building is equipped with heterogeneous devices (sensors like thermometer, presence sensor, luminosity sensor, power smart meter, smart plug and actuators like air conditioner, lamp and dishwasher).

For this case study an application is developed. In the Smart Building application (SBA) each device is presented by a software component. Such software components include: a building manager component, a lamp component, an air conditioner component, a thermometer component, a dishwasher component, a photovoltaic panel component, a power smart meter component, a smart plug component, a presence sensor component, a luminosity sensor component and a user agent component. When the device has enough capability, the software component can be deployed on it. Otherwise, the software component is deployed on the controller.

The principal software component is the Building Manager, deployed in the Controller alongside the other software components. The Building Manager manages the functioning of the building. The software components of the Smart Building application are organized into communication sessions. Each session includes a group of components communicating in order to achieve a specific objective. The components of the Smart Building application interact and communicate through the Channel Manager (CM). Each session of the SBA is managed by a CM component.

6. Deployment approach for Smart Building

In order to ensure the deployment of some software components of the SBA in various machines, we need a Deployment tool. In our work, we used a deployment service of the FACUS framework [14]. We used also the OSGI framework which includes components called *bundles*. Bundles can be remotely installed, started, stopped, updated and uninstalled without requiring a reboot of the OSGi framework. For this reason, we choose to implement the Channel Managers components as OSGi bundle. This service performs the deployment process using three entities: The Deployment Manager, the Components Repository and the Deployment Agent.

To prepare for the deployment process of the SBA and deciding the actions to be taken during the deployment process in order to successfully execute the deployment process, three questions have to be answered:

- *What is deployed?* The first question concerns the nature of the deployed software and its components.
  Answer: The Smart Building application is a component based application. It is composed of several software components. The components that will be deployed are the Channel Manager and the software component of the Smart Building application which are : Building manager, air conditioner, lamp, dishwasher, smart plug, thermometer, photovoltaic panel, luminosity sensor, presence sensor, power smart meter, user agent.
- *Where is the software deployed?* The second question is about the location where the software components of the application will be deployed.
  Answer: The Smart Building software components are deployed in different machines that have enough computing capabilities and connected through the network.
- *How is the deployment performed?* The last question relates to the realization of the deployment process: the description of the constraints, the organization and the architecture of the deployment descriptor.

Answer: The administrator of the Smart Building application defines the list of constraints. The administrator knows from the beginning the architecture of the building, the number of the devices in the building where the software components will be deployed in order to organize the deployment descriptor.

6.1. Deployment domain and deployment descriptor

The deployment of a software is done on several consumer sites. All of these sites constitute the ≪deployment domain≫ where the software components are deployed, and on which the deployment of the software components is described by the administrator of the application called ≪deployment descriptor≫.

- Deployment domain : The deployment domain is the set of networked machines or devices (consumer sites) where the software components are deployed.
- Deployment descriptor : The deployment descriptor is a mapping between a software component and the deployment domain (networked machines). It specified for each software components the location where it will be deployed, the action that will be executed on each software components (deploy, install, start, etc.).

6.2. Deployment constraints

In the Smart Building application, the administrator defines a set of constraints. It specifies constraints related to the devices resources such as the available memory of the machines where the software components are deployed, energy constraints which are related to the energy consumption in the building and it defines a set of flexibility values for lamps and air conditioners. The flexibility value for lamps is the value that the Building Manager can subtract from the intensity value of lamps when it executes the advanced energy saving strategy. While the flexibility value for air conditioners is the value that the Smart Building Manager can add to the temperature value of air conditioners when it executes the advanced energy saving strategy.

The administrator of the Smart Building defines for each parameter (used memory, energy consumption) a threshold value. Those thresholds must be respected while deploying the software components of the application.

6.3. Deployment activities of the Smart Building application

Based on the sequences of activities earlier mentioned, we have performed a set of activities in order to deploy the Smart Building application:

- Release: this activity is done by the producer of the Smart Building application, where he/she prepares the different software components of the application (Building manager, lamp, air conditioner, etc.) and assembling them into packages, those packages will be deployed later on one or more target sites.
- Installation: this activity refers to the installation of the software components in the target sites. This activity is done by the administrator of the application. This activity is divided into steps done manually such as the JAR file of each software components

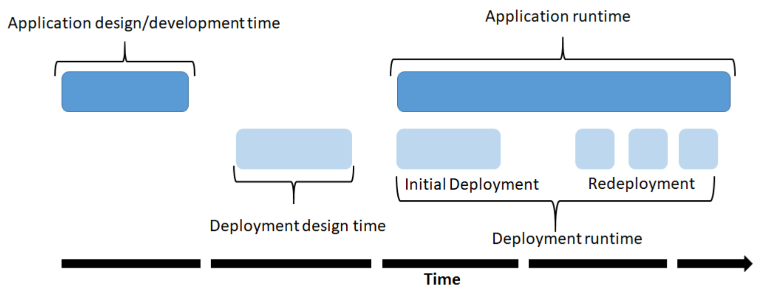

Figure 3: Deployment timeline

which must be placed in the targeted devices, and steps done automatically such as the deployment of the Channel Managers. In order to achieve the steps done automatically, the administrator prepares the deployment descriptor.

- Activation: following the preparation of the deployment plan, the application in this step can be launched.
- Adaptation: the Smart Building application is able to adapt itself to dynamic changes. Those changes may be resulted from a user requirement, environmental changes (such as over consumption of the energy in the building).
- Redeployment: The redeployment activity may for example be required during an excessive use of the machine memory. This activity consists of moving some components from one machine to other while preserving the functionality of the application.

6.4. Deployment timeline

Figure 3 presents the deployment timeline related to the preparation of the deployment process and the execution of the deployed application. Deployment activities take place before and during the execution of the application. Before running, the application is installed and activated: this sequence of activities is commonly named **initial deployment**. During the execution of the application, environmental changes may happen. Thus, adaptation and redeployment activities take place in the deployment process: this sequence of activities is commonly named **redeployment**. Deployment runtime is the period when the deployment descriptor is running.

6.4.1. Initial deployment

In order to deploy the software components of the Smart Building application in different devices, we use the deployment service as we mention previously.

The deployment service takes a deployment descriptor (which is prepared by the administrator of the application) as input. This file defines what component will be deployed on each machine. By receiving the deployment descriptor, the deployment service identifies the required software components that must be installed on each device (CM and the software components). Then, it connects to the deployment agents which are installed in various machines, informing them about all needed components.

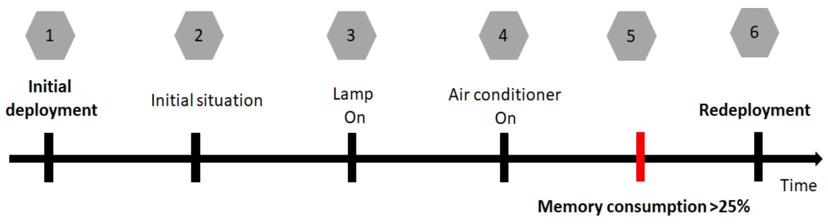

Figure 4: Execution of the deployment process scenario of the Smart Building application

6.4.2. Redeployment

The Deployment Service supports the redeployment at runtime enabling the Smart Building application to adapt itself in order to handle the changes that have been occurring during the application execution.

The redeployment process is executing when one of the networked machine is in a critical state. A machine is considered in a critical state when its resources are less than the threshold values that the administrator of the application defined. Once a machine is in a critical state, an adaptation query is generated by the decision module [15, 16]. This module interact with the application ensuring the selection of the most suitable redeployment of the CM components.

7. A deployment scenario of the Smart Building application

Our deployment process can be applied to many Smart Buildings examples. In order to evaluate it, we consider a Smart Building example that is structured as follows:

- A bedroom contains: a lamp, an air conditioner, a presence sensor and a luminosity sensor.
- A living room contains: two lamps, two air conditioners, two presence sensors and a luminosity sensor.
- A kitchen contains: two lamps, a dishwasher, two presence sensors and two luminosity sensors.
- A bathroom contains: a lamp, a presence sensor and a luminosity sensor.

The Figure 4 presents a deployment scenario of the Smart Building application. It includes the sequence of events which appear in the building over the time. The horizontal axis in this figure represents time, and the different events are marked from (1) to (6).

1 : The first event (1) in the time axis represent the initial deployment which includes under tasks which are:

- The administrator of the SBA defines a set of constraints which must be respected during the execution of the Smart Building application. In our work, we are interested to the constraints related to the resources of machines like the memory consumption of the machines. The administrator defined a threshold value for the memory consumption (25%).

- The administrator makes a correspondence between the software components of the application and the device where the component will be deployed on, and prepare the deployment descriptor.
- When the deployment descriptor is prepared, the administrator executes the Building manager and the initial deployment is starting. Afterwards, the channel managers are deployed in the controller and the software components are deployed in the targeted devices.

2 : The second event (2) represents the initial situation of our deployment process, where all the devices are turned off.

3 : The third event (3) is the arrival of the user requirements. The user turn on a lamp with an intensity equal to 1000 Lumen through its device interface.

4 : The fourth event (4) the user turns on an air conditioner with temperature value equal to 25C through its device interface.

5 : While communicating with the Building Manager, the memory consumption in the Controller increases. The Building manager detects that the Controller is in critical state, it sends an adaptation request to the Decision module. This module is responsible of choosing the most suitable redeployment of the CM components. Then this module sends the new deployment descriptor to the Building Manager. The Building Manager sends the deployment descriptor to the Deployment Service which indicates for each deployment Agent the components that it must deploy in each device.

6 : The Building Manager receives the new deployment descriptor the redeployment in executed automatically. The deployment service receives the deployment descriptor and connects to the deployment agents indicating the new location of the CM components.

8. conclusion and future work

In this paper, we propose a deployment approach for the Smart Building application. First, we present tools that ensure the deployment of software applications. Then, we briefly present the case study Smart Building. After that, we present our deployment approach for the SBA. Finally, we apply the deployment approach on a Smart Building example using the deployment service which ensures the deployment of some software components of the application in a set of machines. Also, the deployment service supports the redeployment at runtime enabling the SBA to adapt itself in order to handle the changes that have been occurring during the execution of the application.

As future work, we aim to evaluate our approach through real world experiments.

Acknowledgments

This work was partially supported by the LABEX-TA project MeFoGL:"Méhodes Formelles pour le Génie Logiciel".

References

[1] A.H. Buckman M. Mayfield Stephen B.M. Beck, “What is a Smart Building", Smart and Sustainable Built Environment, vol. 3, pp.92–109, September 2014.
[2] P. Ehrlich, “What is an intelligent building?", Building Intelligence Group, 2007.
[3] G.D. Abowd, “Classroom 2000: An experiment with the instrumentation of a living educational environment", IBM Systems Journal, 1999.
[4] W. Haijiang L.J. KwokWai S. Wang, “Intelligent building research: a review, In Software Engineering Research", Management and Applications, pp. 143–159, 2005.
[5] Dounis A.I. Wang Z., Wang L. and Yang R., “Integration of plug-in hybrid electric vehicles into energy and comfort management for smart building", Smart Buildings and Demand Response , 260–266, 2012.
[6] A. Heydarnoori, “Deploying component based applications: Tools and techniques", In Software Engineering Research, Management and Applications, 2008.
[7] R. Boujbel J.P Arcangeli and S. Leriche, “Automatic deployment of distributed software systems: Definitions and state of the art", Journal of Systems and Software (2015), 198–218.
[8] InstallShield Developer, https://www.installshield.com.
[9] Microsoft Windows Installer, https://www.microsoft.com, 2006.
[10] A. Flissi J. Dubus N. Dolet P. Merle, “Deploying on the Grid with DeployWare", 2008.
[11] P. Goldsack J. Guijarro S. Loughran A. Coles A. Farrell A. Lain P. Murray P. Toft, “The SmartFrog Configuration Management Framework", Operating Systems Review , pp 16–25, 2009.
[12] OSGi Alliance, “Osgi service platform core specification", http://www.osgi.org/Specifications, 2007.
[13] C. Louberry P. Roose M. Dalmau, “Kalimucho: Contextual Deployment for QoS Management", Distributed Applications and Interoperable Systems, 2011.
[14] G. Sancho, “Adaptation d’architectures logicielles collaboratives dans les environnements ubiquitaires. Contribution à l’interopérabilité par la sémantique ", Université des Sciences Sociales - Toulouse I , 2010, PhD Theses.
[15] I. Abdennadher , I. Bouassida Rodriguez , M. Jmaiel, “A Design Guideline for Adaptation Decisions in the Autonomic Loop", Procedia Computer Science, 2017.
[16] I. Abdennadher , “DAACS : a Decision Approach for Autonomic Computing Systems", Journal of supercomputing, 2021.