Information Modeling for Virtual and Augmented Reality

Original
Information Modeling for Virtual and Augmented Reality / Patti, Edoardo; Mollame, Angelo; Erba, David; Dalmasso, Daniele; Osello, Anna; Macii, Enrico; Acquaviva, Andrea. - In: IT PROFESSIONAL. - ISSN 1520-9202. - 19:3(2017), pp. 52-60. [10.1109/MITP.2017.43]

Availability:
This version is available at: 11583/2642720 since: 2017-07-10T13:04:10Z

Publisher:
IEEE

Published
DOI:10.1109/MITP.2017.43

Smart Systems

Information Modeling for Virtual and Augmented Reality

Edoardo Patti, Angelo Mollame, David Erba, Daniele Dalmasso, Anna Osello, Enrico Macii, and Andrea Acquaviva, Politecnico di Torino, Italy

The authors combine building information modeling (BIM) data with ambient information from devices deployed in smart buildings. Their Android-based application can then offer environmental building information integrated with BIM data in an augmented and virtual reality environment.

Today's buildings are equipped with various types of sensor nodes—either wired or wirelessly connected to a home or building network—for monitoring and management purposes. These devices provide ambient information about environmental and energy-related parameters to increase awareness in building occupants and managers, and to facilitate feedback actions or plan interventions. Nevertheless, how to enable interoperability across devices adopting heterogeneous communication protocols or standards is the object of intensive research. Service-oriented middleware technologies attempt to address this issue.[1,2]

In the context of smart buildings, effectively engaging users and exploiting so much potential information requires contextualizing physical data from sensors by combining it with other types of ambient information, such as building characteristics, topology, and infrastructures. To this end, building and energy managers can exploit *building information modeling* (BIM),[3] a methodology that provides an accurate virtual model, usually in 3D, of a building and its infrastructures. This model is created, updated, and consulted during the design, development, and management of the building itself.

Here, we present a methodology and an associated Android-based mobile application that integrates sensor data with building models, allowing end users (that is, technicians and building and energy managers) to navigate in a virtual or augmented building environment and access context-related physical environmental parameters such as temperature, humidity, and energy consumption. The values of these parameters can be easily correlated with other building characteristics and infrastructural properties, such as gas, electricity, or heating networks.

Our methodology leverages a distributed software architecture composed of underlying middleware services to access sensor data in a hardware-independent way and combine it with BIM models interactively.

Our work's main aims are to

- collect environmental information from heterogeneous devices via a distributed software architecture;
- combine BIM structural data with real-time information to improve building maintenance;
- improve the visualization of such integrated information in a virtual and augmented reality environment;
- move BIM from desktop computers to mobile devices by providing an innovative, portable tool for building and energy managers;
- provide end users with a tool for interacting with the building to access heterogeneous data available from multiple pervasive sources; and
- increase user awareness about energy consumption and environmental conditions.

Augmented reality (AR) has been defined as the link between the real world and virtual reality (VR),[4] which in turn is a completely artificial environment for simulating reality. Each real object intrinsically provides a significant amount of information that sometimes is not immediately perceived by users. AR aims to make such information visible by overlapping digital reality with physical objects.

The recent evolution of smart buildings provides new horizons for VR and AR application development, which could make building management easier and increase user awareness about energy consumption. Our proposed Android application exploits both AR and VR to provide building information, overcoming limits related to 2D visualization. Indeed, it presents a 3D environment in which real-time building information, coming from pervasive devices, is combined with structural and architectural data provided by BIM.

Contributions in Building Information Modeling

In the field of building design and maintenance, BIM[3] is a powerful development and management methodology for architects, engineers, and designers to achieve an accurate virtual model of a building, often called a *proto-building*.[5] BIM goes beyond geometrical knowledge of the building to include all the essential information for its design, construction, and management. This approach changes the relationship among the different skills involved in the building process, providing continuous correspondence between computer modeling and construction. Hence, the digital model is not a simple drawing, but rather becomes the starting point for a multitude of possible representations.

Nowadays, our buildings are already equipped with heterogeneous devices to monitor and manage environmental parameters such as air temperature, relative humidity, and power consumption. Such heterogeneous devices can be either wireless or wired, and they exploit protocols such as ZigBee, EnOcean, or BACnet. To enable interoperability, distributed software architectures have to be considered. The recent development of middleware and service-oriented architectures seems to be promising in this direction.[1] Indeed, middleware technologies should implement the abstraction software layer required to achieve interoperability across heterogeneous devices, including existing building management systems (BMSs). Our work puts this concept in operation to enrich and combine BIM with real-time building energy consumption data, providing a mobile application suitable for Android tablets.

To increase awareness about building energy consumption and to make maintenance work easier, our proposed application leverages a distributed, event-based, service-oriented infrastructure that exploits Smart Energy Efficient Middleware for Public Spaces (SEEMPubS).[2,6] It lets end users interact with the system to access building information available from multiple pervasive sources. With respect to other solutions, it mixes structural information with fine-grained energy and environmental data coming from heterogeneous wireless or wired pervasive devices. It provides an innovative tool for the BIM process that exploits both VR and AR to provide real-time data that is also related to structural information about monitored environments. Usually, AR exploits markers for linking real-world objects to digital enhancements (such as video, animations, or interactive graphs) that are accessed via tablet or smartphone. In this work, Quick Response (QR) codes have been adopted as markers. Hence, our proposed solution aims at moving forward from a private household or single office perspective to the larger one of buildings and public spaces. Furthermore, our application breaks the constraints imposed by 2D visualization, moving toward 3D visualization of building environments.

Smart Building Management Architecture

In the context of the Internet of Things (IoT) and ubiquitous computing, one stumbling block concerns the coexistence of several different technologies. In addition, enabling interoperability across pervasive and heterogeneous devices, both wireless and wired, is the key issue to turning buildings into smart buildings. For this purpose, middleware technologies can be exploited, easing the development of smart building applications.

Although efforts toward the interoperability of different building automation technologies already exist, creating integrated systems for different IoT technologies is still a cumbersome task. Furthermore, how to deal with interactions with technologies beyond the scope of building automation remains a challenge. A common approach to solving such interoperability issues consists of exploiting middleware technologies.

In the following, we present our distributed middleware-based building monitoring and management system.

Middleware for Smart Buildings

To enable interoperability among heterogeneous IoT devices, the proposed software infrastructure leverages SEEMPubS,[2,6] extending and improving the functionalities offered by the LinkSmart service-oriented middleware (linksmart.eu/redmine). SEEMPubS comprises a three-layered architecture, with an integration layer, a services layer, and an application layer (Figure 1).

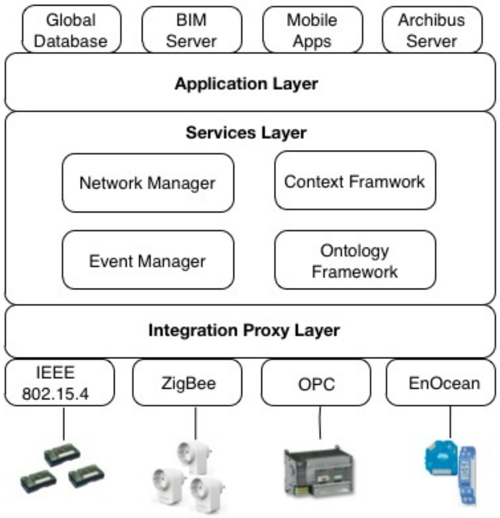


Figure 1. Schema of the Smart Energy Efficient Middleware for Public Spaces (SEEMPubS). This architecture features integration, services, and application layers.

The integration layer represents the integration of heterogeneous devices. It exploits the concept of an *integration proxy* (described later) to enable interoperability across heterogeneous devices by abstracting a certain technology to web services. Hence, the integration layer acts as a bridge between the middleware network and the underlying technology, translating whatever language the low-level technology speaks into web services.

The services layer is the core of the SEEMPubS middleware. It is designed for developing smart building applications by providing developers with a set of components that expose functionalities as Web services. Its main components are as follows.

The *network manager* establishes a peer-to-peer (P2P) network, allowing direct communication among all the applications in the middleware network. Web service calls are routed through the network manager, which creates a SOAP tunnel to the requested service endpoint regardless of whether peers are behind a firewall or network address translation (NAT).

The *event manager* provides for event-based communication based on a publish/subscribe service[7] for SEEMPubS web services. This enables the development of loosely coupled, event-based systems and increases scalability. Indeed, the publish/subscribe approach removes interdependencies between information producers and consumers. In smart buildings, we deal a lot with events coming from both devices and distributed software. Hence, this mechanism is a key requirement for developing systems and applications.

Context and ontology frameworks together manage semantic knowledge about the application domain and the implemented system. This includes metadata about sensors and actuators and also their relation to domain model objects, such as appliances, buildings, and rooms. Finally, these frameworks provide developers with web services to query any kind of information from a rich domain model (for instance, the location or capabilities of a sensor, a list of all sensors in a room, or an actuator with a certain control capability).

Finally, the application layer—the top layer in the proposed architecture—provides a set of tools for using the integrated systems and information that is available. Furthermore, it makes available a set of APIs and web services for developing distributed, event-based applications to manage buildings and post-process data coming from the lower layers.

Integration Proxy

The integration layer is in charge of enabling interoperability between heterogeneous devices, as shown in Figure 1. It is based on the concept of an integration proxy, which is a middleware-based software component. The integration proxy abstracts the underlying heterogeneous IoT technologies, either wired or wireless, and enables interoperability among them. It provides the following main functionalities:

- integrating such devices in the proposed infrastructure via web services, through which sensor data are read and can be used for visualization or for energy management policies;

- collecting environmental data from sensor nodes into a local database that preserves them from network failures and can be accessed asynchronously;
- pushing environmental data into the infrastructure via an event-based approach thanks to the event manager;
- allowing remote reconfiguration of sensor node parameters, such as the sampling rate of physical quantities; and
- allowing remote control of actuator devices.

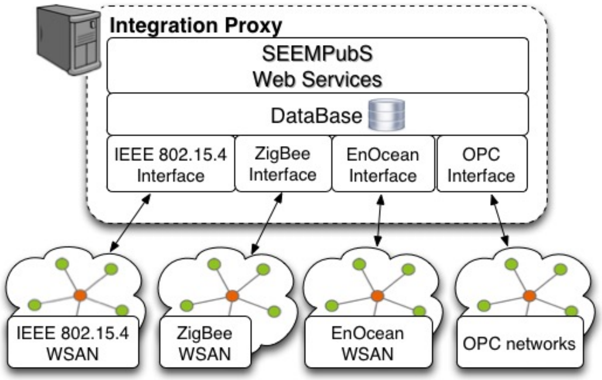


Figure 2. Integration proxy. The proxy consists of three sublayers, runs on a PC, and communicates directly with the underlying heterogeneous technologies.

As Figure 2 shows, the integration proxy consists of three sublayers. It runs on a PC and communicates directly with the underlying heterogeneous technologies. The dedicated interface represents the lowest sublayer and receives all the information coming from the devices, regardless of the chosen communication protocols, hardware, or network topology. Hence, each technology needs a specific software interface, which interprets the environmental information and stores it with its time stamp in the integrated database. Because data are locally stored, the local database makes the whole infrastructure flexible and reliable with respect to backbone network failures. The SEEMPubS sublayer interfaces the device networks to the rest of the infrastructure, abstracting the functionalities of the underlying technologies into web services. This enables interoperability between heterogeneous devices and eases their remote management and control. At this layer, the environmental data is immediately sent to the middleware network, exploiting the publish/subscribe approach provided by the event manager. Hence, other middleware applications, such as the *global database* (GDB, described later), can receive and post-process it. In particular, we developed integration proxies to manage wireless sensor and actuator networks (WSANs) that exploit IEEE 802.15.4, ZigBee, and EnOcean protocol stacks. Furthermore, to provide backward compatibility with wired technologies, we developed a specific integration proxy to enable interoperability with the OPC Unified Architecture (opcfoundation.org), which incorporates all the functionalities provided by different standards, such as BACnet.

Global Database Server

Due to hardware limitations (such as limited radio range for wireless devices), a single integration proxy might not be enough to monitor and manage large building complexes. Consequently, it might not be able to interact with all the IoT devices deployed in the whole building. Hence, multiple integration proxies could be needed. We developed and introduced into the proposed software infrastructure the GDB, which lies in the application layer. The GDB is a middleware-enabled software that implements a database to merge and synchronize all the information about the deployed devices' configurations and real-time environmental data coming from the integration proxies. In addition, GDB exposes dedicated web services to provide a centralized entry point for accessing this information to other middleware-based applications, such as the Android application we describe later.

The GDB, during its bootstrap, requires the *context manager* to provide a full list of the devices deployed in each room of the building. Afterward, it subscribes to the event manager to receive environmental data sent by these devices through the integration proxies. This process is periodically repeated to keep the GDB updated so that, if a new end node is deployed, only the ontology managed by the context manager needs to be updated. Then, this change will be automatically propagated to all the other middleware-based software, such as the GDB. Note that these updates can be done without restarting the whole software infrastructure. The propagation of environmental information from integration proxies to the GDB exploits the publish/subscribe approach provided by the event manager. However, in case of backbone network failures, when the Internet connection is restored, the integration proxies automatically synchronize their information thanks to the local database integrated in each integration proxy.

The GDB integrates and also works as a BIM server implementing a communication protocol for uploading new 3D models into the system, and for their download by the Android application. The BIM server also handles the validation of each new model uploaded into the system by guaranteeing that the information it contains (namely, the references to the rooms and sensors that the 3D model represents) is consistent with the information stored in the GDB.

Furthermore, thanks to the web services programming paradigm, the GDB integrates the third-party Archibus Building Facilities Management software (www.archibus.com) into the proposed infrastructure and merges its information with the real-time building data and 3D parametric models. Thus, when the Android application requests information for a specific room, the GDB sends the 3D parametric model, the related information retrieved from Archibus, and all the environmental data collected by the devices deployed in that room.

Software Infrastructure

Figure 3 outlines our proposed infrastructure. As introduced previously, SEEMPubS enables a P2P network, thanks to the network manager, creating a SOAP tunnel to overstep firewalls and NAT. To exploit the functionalities provided by the network manager, each peer needs to integrate it or to call its web services remotely. In the latter case, we deployed a network manager as gateway (NMGW) to provide an entry point to the middleware network for all the applications that do not integrate the network manager (for instance, devices with limited resources). To be reachable, the NMGW needs to run in a computer or server with a public IP address. The other peers of this infrastructure are the integration proxies, the event manager, the context manager, and the GDB that implements the BIM server and integrates Archibus in the middleware network.

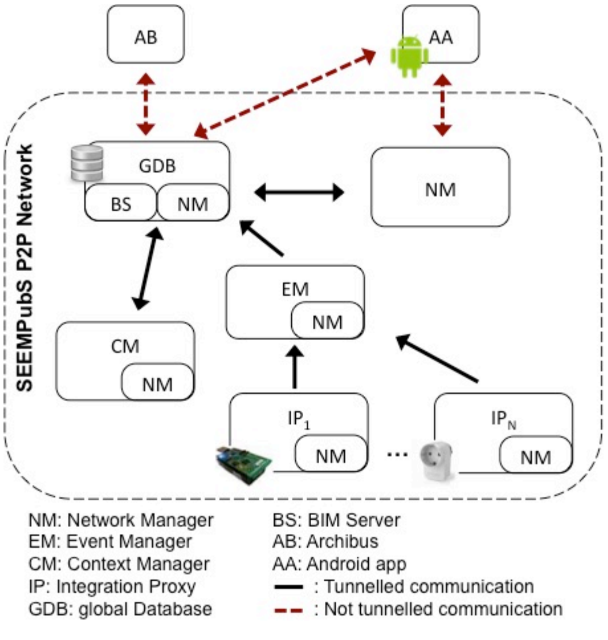


Figure 3. Software infrastructure. We can see the main components of the proposed solution.

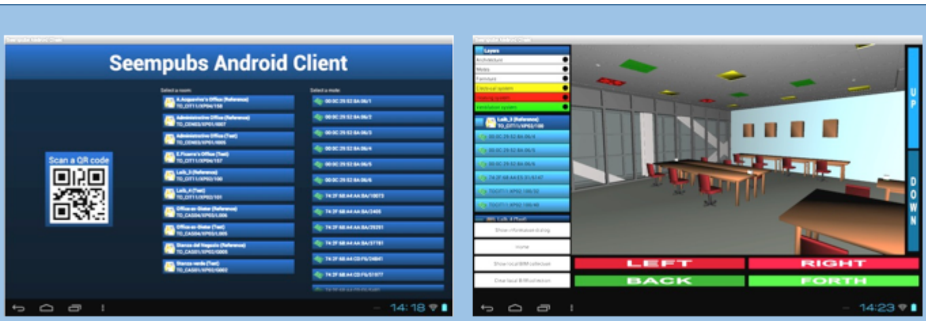
In addition, Figure 3 shows the communication flows between each peer in the P2P network. In brief, each device sends environmental data to the integration proxy, which publishes it as a new event to the event manager. This new event is delivered to all the applications subscribed to it, including the GDB. As described, the GDB requests the list of deployed devices from the context manager first, then subscribes to the event manager to receive data from these devices. Finally, our Android application joins the middleware network by exploiting the NMGW that routes and tunnels its web service calls to the GDB, which replies with architectural and environmental data. As an alternative, if the server, where the GDB runs, is configured with a public IP address, the Android app can directly call its web services without NMGW support. Note that the GDB replies, providing structural information coming from Archibus as well.

Android Application

To increase user awareness and enhance any possible maintenance work in the building, we've developed an Android application to provide information on the building that exploits both AR and VR. This solution aims to overcome the limits related to 2D visualization, presenting a BIM 3D model enriched with real-time environmental and energy-related information. The application provides three main functionalities:

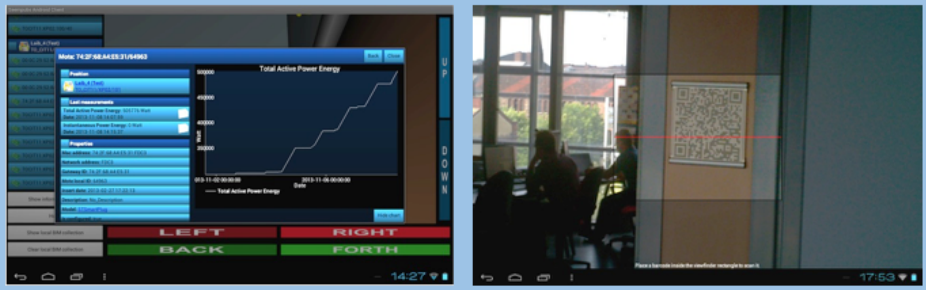
- an interactive interface for visualizing real-time monitoring information about environmental conditions and energy consumption retrieved from the remote GDB;
- the ability to combine real-time monitoring information with 3D building models of the monitored room with its internal systems—these 3D models are also provided by the GDB; and
- fast access to location-based information using QR codes to identify rooms and devices before sending information requests to the GDB.

The application has been designed to provide awareness about energy consumption and to be used for maintenance purposes. The identified end users for this application are building and energy managers. In an interesting use case scenario, the monitoring architecture we present collects information about an anomalous environmental condition (such as a rise in the temperature of a room), which is probably related to a failing system (air conditioning). The monitoring system can be programmed to automatically report this fault to the maintenance office, thus greatly reducing response time. The technician, sent to solve the problem, could use the mobile application on a tablet to reach the room, read all the real-time monitoring information available for that room, and get precise information about the location and structure of the failing system, hidden behind the walls, floor, and ceiling. Figure 4a shows the *main activity* in the proposed application, in which the user can choose between VR and AR navigation (an *activity* in Android is a software window that is usually displayed in full-screen mode).

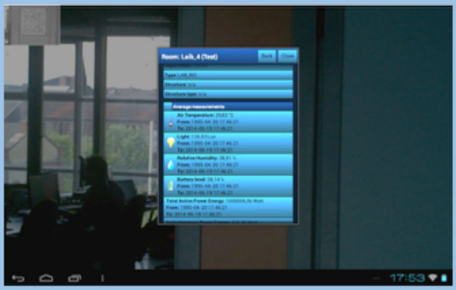


(a) Main Activity (b) VR-Activity: Room View

(c) VR-Activity: System View (d) VR-Activity: Device View



(e) VR-Activity: Real-time and Historical View (f) AR-Activity: Scanning QR Code



(g) AR-Activity: Real-time data View (h) Android Applicartion use case video

Figure 4. An open-plan office with building information management and real-time information, visualized with the Android application. (a) In the main activity, the user can choose between virtual reality (VR) or augmented reality (AR) navigation. (b) The VR activity merges 3D building models with real-time environmental and energy-related data for devices in the building. (c) Transparency controls let users turn architectural components transparent or invisible to see hidden systems in a room. The info activity is overlapped onto the VR activity to show both (d) real-time information and (e) a timeline of historical environmental data. (f) After a user scans a Query Response (QR) code, the Android application decodes it and requests related real-time information. (g) The application shows the monitoring data overlaid on the video stream through the AR activity. (i) Readers can scan the QR code and click on the resulting link to view a demo of our Android application.

Virtual Reality Navigation

The *VR activity* (Figure 4b) shows the 3D building models and merges them with real-time environmental and energy-related data coming from the devices deployed across the building. From the main activity, the user can select a room or a device. Automatically, thanks to the SEEMPubS web service, the application downloads the corresponding 3D model from the BIM server, which is part of the GDB, and loads it to the VR activity. The application allows the user to navigate in the 3D scene and to selectively change the transparency of different layers. For instance, these transparency controls can be used to turn architectural components transparent or invisible to see the hidden systems in the room, as Figure 4c shows. In the 3D model, the deployed devices are represented as well. When the user selects one of them, an *info activity* is overlapped on the VR activity, showing both real-time information and a timeline of historical environmental data, as shown respectively in Figures 4d and 4e. Moreover, the info activity provides information about the selected device's hardware and configuration settings. All this information is retrieved by calling the SEEMPubS web services exposed by the GDB.

Augmented Reality

To provide information via AR, QR codes have been used as markers. QR code is the trademark for a type of matrix or 2D barcode. QR codes are now widely used for their fast readability and large storage capacity; indeed, they can encode numbers, strings, and binary data. In this work, we used them to encode a room ID to identify a unique room in the system or a device ID to identify a unique device in the system.

In this scenario, the user scans the QR code first (Figure 4f); then, the Android application decodes it and requests the related real-time information by calling the GDB's web services. Finally, the application shows the monitoring data overlaid on the video stream through the *AR activity*, as shown in Figure 4g. When a QR code associated with a deployed device is successfully scanned, the AR activity shows the sensed environmental data, configuration parameters, and the hardware information of the device itself. With a QR code related to a room, the AR activity shows the information collected by the deployed sensors in the room, such as air temperature, relative humidity, illuminance, and energy power consumption.

Note that the room's information, given by the info activity, VR activity, and AR activity, also comes from Archibus.

A video demo of our Android application is available on YouTube: just scan the QR code in Figure 4h and click on the resulting link (www.youtube.com/watch?v=nyN_uJmy_tc).

3D Model: From Revit to Android

In this section, we describe our methodology for moving the visualization of BIM 3D models from the desktop computer domain to mobile devices, such as tablets equipped with Android OS. Typically, these devices have limited computational resources with respect to a modern desktop computer; hence, to show the 3D building models, we adopted the Java Multimedia File (JMF) format. JMF is an optimized format suitable for the Java environment, which is the basis of Google's Android software development kit. Indeed, JMF files can be read and written very efficiently, without any conversion, computation, or temporary allocation. Before obtaining the final JMF model, several steps must be followed, as Figure 5 shows.

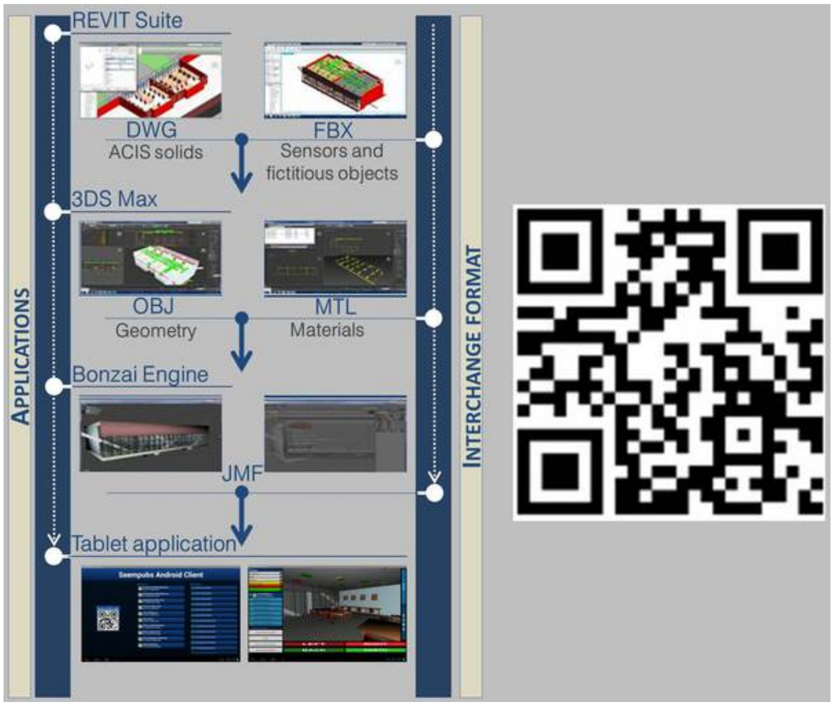


Figure 5. Procedure to obtain a 3D model suitable for Android tablets. The main steps include (from top to bottom) modeling with Revit; exporting from Revit to 3DS Max; exporting from 3DS Max to the Bonzai engine; and visualizing on Android tablets.

We started from the as-built documentations, verified—if necessary—with specific surveys. As-built documentations consist of the original blueprints revised to reflect any changes made during the building process. The building was split into different parts: architecture, and electrical, heating, ventilation, plumbing, sewage, and gas systems. Each part was drawn as a separate workset, and then all the worksets were merged in the final model. Furthermore, two additional worksets have been shaped: the first represents the furniture, whereas the second models the deployed devices, adding their IDs as metadata. These device IDs will be the input for the GDB's web services to retrieve environmental information.

First, we used Autodesk Revit 2013 (www.autodesk.com/products/revit-family/overview) to make the 3D models. Then, they were exported to two exchange formats: DraWinG (DWG) and, only for the device workset, FilmBox (FBX). FBX allowed us to export the geometry of the model while keeping the IDs of different objects unchanged. The DWG format, on the other hand, let us solve some issues related to the complexity of model geometries. Indeed, DWG lets us choose which type of solid to export and, thanks to ACIS, to reduce the amount of polygons forming the model's surfaces. Then, these files were imported into 3DS Max (www.autodesk.com/products/3ds-max/overview) to convert them into two other formats: OBJect (OBJ), for storing the models' geometries, and Material Template Library (MTL), which contains the description of the materials. The import operation was carried out connecting the different exported files via Manage Link Command. Finally, the OBJ and MTL files were inputs for the Bonzai Engine tool (bonzaiengine.com/index.php) to obtain the JMF model that can be subsequently uploaded in the GDB's BIM server and sent to the Android application if requested by end users.

Impacts on Building Management

In this section, we analyze the benefits and the impacts that this solution offers in building management. The presented software infrastructure can be considered as a common repository for building information, ensuring data uniqueness among different actors, and hence avoiding data inconsistency. This can be considered a key requirement to be addressed. Indeed, during building design, development, and management, different actors are involved (designers, energy managers, facility managers, and so on) who need to exploit the same updated information.

Furthermore, both the software infrastructure and the Android application are innovative tools for building and energy managers to visualize and manage information—also in (near) real-time—about the building's status. We proposed a methodology for moving BIM models from the domain of desktop computer to mobile devices. Hence, users can easily access such information from anywhere in the building, improving on-site interventions and management. This is crucial, especially for energy and building operators who have to manage big buildings or blocks of buildings, such as university campuses.

Finally, most of today's building blueprints are based on 2D CADs. Hence, exploiting 3D BIM models, enriched with (near) real-time environmental and energy-related data, can have a positive and innovative impact on training technical staff by exploiting both VR and AR jointly with emerging mobile technologies, such as tablets or Oculus Rift.

In the field of building management, in which 2D maps are still used to represent rooms and systems, our proposed solution aims to overcome the limits of 2D visualization by providing a 3D building model visualization environment. Furthermore, the proposed solution allows users to access heterogeneous information from different pervasive sources in a single building management context in which all the available information is integrated. Finally, it appears to be a significant step forward in the field of BIM thanks to its heterogeneous, real-time data integration.

References

1. C. Warmer et al., "Web Services for Integration of Smart houses in the Smart Grid," *Proc. Grid-Interop Forum*, 2009, pp. 1–5.
2. E. Patti et al., "Event-Driven User-Centric Middleware for Energy-Efficient Buildings and Public Spaces," *IEEE Systems J.*, vol. 10, no. 3, 2016, pp. 1137–1146; doi: 10.1109/JSYST.2014.2302750.
3. C. Eastman et al., *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors*, John Wiley & Sons, 2008.
4. P. Milgram et al., "Augmented Reality: A Class of Displays on the Reality-Virtuality Continuum," *Proc. SPIE: Telemanipulator and Telepresence Technologies*, vol. 2351, 1995; doi:10.1117/12.197321.
5. J. Tobin, "Proto-Building: To BIM Is to Build," *AECbytes* newsletter, May 2008.
6. A. Osello et al., "Energy Saving in Existing Buildings By an Intelligent Use of Interoperable ICTs," *Energy Efficiency*, vol. 6, no. 4, 2013, pp. 707–723.
7. P.T. Eugster et al., "The Many Faces of Publish/Subscribe," *J. ACM Computing Surveys*, vol. 35, no. 2, 2003, pp. 114–131.

Edoardo Patti *is a postdoctoral research fellow at Politecnico di Torino, Italy. His research interests are in ubiquitous computing, the Internet of Things, smart systems and cities, middleware solutions, and software architecture for ambient intelligence. Patti received a PhD in computer engineering from Politecnico di Torino. Contact him at edoardo.patti@polito.it.*

Angelo Mollame *is a software development engineer in test at Microsoft in Washington. His research interests include virtual and augmented reality, and smart systems and cities. Contact him at angelo.mollame@gmail.com.*

David Erba *is a research fellow at Politecnico di Torino, Italy. His research interests are in design and building information modeling. Contact him at david.erba@polito.it.*

Daniele Dalmasso *is a postdoctoral research fellow at Politecnico di Torino, Italy. His research interests are in design and building information modeling. Contact him at daniele.dalmasso@polito.it.*

Anna Osello *is an associate professor in at Politecnico di Torino, and head of the "Drawing to the Future" research group. Her research interests include design and building information modeling. Osello received a PhD in drawing and survey of building heritage from La Sapienza University of Rome. Contact her at anna.osello@polito.it.*

Enrico Macii *is a full professor in the Department of Control and Computer Engineering at Politecnico di Torino, Italy.*

His research interests are in the design automation of digital circuits and systems, with particular emphasis on low-power design aspects. Macii received a PhD in computer engineering from Politecnico di Torino. Contact him at enrico.macii@polito.it.

Andrea Acquaviva *is an associate professor in the Department of Control and Computer Engineering at Politecnico di Torino, Italy. His research interests focus on parallel computing for distributed embedded systems such as multicore platforms and sensor networks, and simulation and analysis of biological systems using parallel architectures. Acquaviva received a PhD in electrical engineering from Bologna University. Contact him at andrea.acquaviva@polito.it.*

Smart building is a rising interdisciplinary research field that aims to improve the monitoring, management, and maintenance of buildings. The authors present an innovative solution for combining building information modeling (BIM) data with ambient information collected by heterogeneous devices deployed in the building. To collect environmental information, they exploit a distributed software architecture. It enables interoperability between heterogeneous data sources—either physical devices, such as sensor nodes, or third-party software such as Archibus—where building information resides. On top of this infrastructure, they developed an Android-based application that presents environmental building information integrated with BIM data in an augmented and virtual reality environment. The proposed solution gives users awareness about building conditions and energy consumption.

smart building, middleware, Internet of Things, distributed systems, virtual reality, augmented reality, BIM, building management, pervasive computing