# Easy Provisioning of Building Automation Systems using Visual Light Communication

Lukas Krammer, Daniel Lechner, Felix Knorr, Stephan Cejka
Siemens AG, Corporate Technology
Email: [lukas.krammer, daniel.dl.lechner, felix.knorr, stephan.cejka]@siemens.com

***Abstract*—In the past decades building automation and control systems emerged from simple control systems for heating, ventilation and air conditioning facilities to complex applications covering different domains and controlling a huge number of entities. When considering complex automation systems, the whole life-cycle has to be taken into account. Besides operation, the installation and configuration is of utmost importance. These initial steps of a building automation systems require a huge amount of manual engineering and are often the source of errors.**

**In order to reduce engineering effort and error probability, this paper introduces a new provisioning approach for building automation systems based on visual light communication. This concept significantly simplifies the identification of technical components with almost no additional hardware requirements. Technical components are communicating with a mobile device in order to allow an easy and intuitive identification and further configuration of the system. In order to show the feasibility of this approach, the concept is integrated in typical engineering processes. An implementation based on a state-of-the-art building automation technology proves the practical applicability of the concept.**

## I. Introduction

Building Automation (BA) Systems are complex systems covering domains such as heating, ventilation and air conditioning (HVAC) as well as lighting and shading. Safety and security applications such as fire safety systems or surveillance systems are further domains within BA.

In contrast to other automation domains (e.g., industrial automation), there is a prevalent cost pressure in building automation, since the automation system is a cost factor in the life cycle of a building that cannot be neglected. Besides costs for the operation of an automation system, significant effort is necessary for installation and configuration. Additionally, reconstruction and modifications of a building cause efforts for adapting the automation system as well. Due to increasing complexity of automation systems, the increasing number of components in the system and rising sizes of buildings, efforts for installation, configuration and reconfiguration are getting more and more crucial.

While the functional binding of sensors and actuators can be done or supported by semantic technologies [1], installation and identification still remains a mostly manual task which causes huge effort and expensive configuration errors. In order to improve this, some engineering tools allow an identification of devices by using (bar-)codes which are mounted on the devices. These codes can be read by dedicated scanners which are connected to the engineering tool – this process simplifies the identification of devices. However, if more than one inputs or outputs are connected to a device, the identification of these ”channels” within these devices still has to be done manually.

The approach presented in this paper does not require any tag placed on the device. It uses visual light communication for identifying automation devices. In contrast to other approaches that use visual light as communication medium for data communication (cf., [2]), this approach facilitates visual light communication only during the installation proccess for the identification. Thereby, only unidirectional communication with ultra-low bandwidth is required.

The concept can be used to identify devices as well as single inputs and outputs. Thereby, it is only assumed that devices are equipped with visual indications (e.g., LEDs). This is especially true for room automation components. The visual signal can be detected and processed by a customer-grade mobile device such as a smart phone or a tablet. An application on the mobile device either interacts with an engineering tool by wireless communication (e.g., WiFi, Bluetooth) or even allows a direct binding.

The novel approach does not only allow the identification of physical devices, but it can possibly be used to identify particular in- or outputs as well. For example an intelligent light switch with multiple channels is equipped with LED indicators for each channel. These LEDs can be used to identify the individual channels. Also, the output channels of a switching actuator can be identified automatically when they are controlling lights in a room.

The remainder of the paper is structured as follows: Section II depicts current solutions for provisioning in building automation systems and their problems. In Section III a solution by utilizing visual light communication and augmented reality is presented. A proof of concept showing the feasibility of the solution is introduced in Section IV. Section V concludes the paper with a summary and possible future work.

## II. Current Provisioning of Building Automation Components

### A. Engineering tasks

Engineering, which further subsumes installation, configuration and reconfiguration, can be divided into different subtasks which not necessarily have to be executed in the following order.

- *Installation:* Physical devices have to be installed in a plant. Depending on the physical topology of an automation technology, sensors and actors are either distributed or combined within single controllers.

- *Identification:* Physical devices have to be identified, meaning that a relation between a sensor, an actor or a functional entity and its logical representation has to be established. Depending on the kind of engineering, this can be temporary, dynamic or static. For example, in the basic case, addresses are assigned to inputs or outputs. Alternatively, a complete logical representation of a sensor or actor is created in the data structure of an engineering tool.
- *Binding:* In order to allow realizing arbitrary functions, links among the logical representation of sensors, actors or functional entities have to be established. For example, a relation between a switch and a switching actuator by setting assigning the particular address.
- *Parameterization:* Depending on the particular application, parameters have to be assigned to sensors, actors or functional entities. For example, a boolean function can be applied to a logical function block.

### B. Provisioning

The latter three tasks listed in the previous subsection can be subsumed under the term ”provisioning”. Currently, two solutions for provisioning of devices of typical BA systems such as KNX or LON are used:

*1) Provisioning prior to installation:* It is possible to plan provisioning prior to the installation by addressing and configuring devices according to their future position and function. These devices then need to be installed and operated exactly on the designated position. This causes the following issues:

- An erroneous or insufficient documentation of the planned site - in this domain very usual - leads to massive additional efforts in terms of adaption tasks.
- In bigger complexes the components are necessarily marked, e.g., by using labels, to tell the technician the exact positions. A small mistake can lead to erratic behavior of the whole facility.
- A function test can be done only after the completion of bigger parts, e.g. of a whole floor.

*2) On-site Provisioning:* A second approach is the on-site provisioning, where components are identified in hand with their installation. This is usually achieved by manually pressing a hidden hardware button in combination with the integration into an engineering tool. Strict processes have to be complied with, e.g., making settings within a small interval of time in which starting of a ”programming mode” and the identification of the component in the engineering tool takes place. Main problems of this solution are:

- The responsible person needs to have physical access to all building automation components distributed in the building to press all buttons.
- Every device needs to be set to the programming mode individually. At any time, there must not be two or more devices in this mode. Only after the addressing of a device is finished, the next device can be set to programming mode. Parallel installation in a segment is therefore not possible.
- On some automation systems, during the provisioning phase, no or only very restricted communication on the communication channel is allowed. This implies that all persons interacting with the automation system have to be informed that they are not allowed to use even basic functionality (e.g., pressing a switch may be prohibited during this phase).
- Depending on characteristics and documentation of installation at the time of provisioning the function of a device is not apparent. For example, it is not evident in the control cubicle which illuminant is controlled by the switching actuator. Naming and addressing is therefore hardly possible in the same step.

### C. Engineering in state of the art technologies

Current automation technologies strongly vary in terms of installation and configuration processes. Basically, it can be distinguished between ad-hoc configuration and a pre-configuration. Some technologies do not even specify engineering processes or tools. Among others, BACnet and KNX are the most representative technologies in the context of building automation [3], [4].

*1) BACnet:* One of the most important standards in building automation is BACnet. However, the BACnet standard only specifies the communication and information representation. The definition of devices is not specified. Even engineering and configuration of devices is left open to manufacturers. Thus, a BACnet based automation system cannot be engineered amongst different manufacturers. Nevertheless, every manufacturer has its own set of engineering tools. A common method of engineering is the use of Continuous Function Charts (CFCs). This method supports a graphical representation of sensors, actuators and function entities and allows graphical binding and parameterization. However, the identification of particular inputs and outputs has to be done manually (e.g., by using schematics, lists or addresses).

*2) KNX:* A further important standard in building automation is KNX, a fully distributed automation technology where sensors and actuators are communicating directly with each other [4]. Beside specifying device functions, communication and information representation, the central component of KNX is its engineering tool called Engineering Tool Software (ETS). This tool can be used to configure and identify devices of different manufacturers. It supports the engineering of even complex systems and provides functions for identification, binding and parameterization of devices. The identification of physical devices is generally done by button presses on the physical device. In case there are more in- or outputs within a device, identification of the particular I/O has to be done manually. In addition to this configuration mode called ”S-Mode” (system mode), there also exists an ad-hoc engineering mode called ”E-Mode”. This engineering is designated for small installations and is based on mutual identification of devices by button presses on the physical devices.

Other automation technologies use similar methods. For example, LonWorks is supported by an engineering tool called LonMaker. EnOcean devices are often configured ad-hoc whereby also manufacturer-specific engineering tools exist.

## III. Provisioning using Visual Lights Communication and Augmented Reality

The aim is to reduce the efforts required for initial provisioning as well as to simplify further engineering. As foundation, visual light communication of components in combination with augmented reality is used. With this technology, automation devices can be uniquely identified by a mobile device. In particular, the automation component emits a unique visual code that can be detected by a mobile application. Depending on the provisioning approach, this identification can either be used to directly bind, e.g., sensors with actuators or link components to a pre-engineered automation scheme.

Typical components of building automation used in rooms already contain such an optical signalization. Control panels for HVAC systems mostly contain of a telltale or displays with an switchable background light. Most of the buttons are equipped with an LED (e.g., for confirmation of an activation). The identification of lighting actuators can also be achieved by identifying the connected light in the room.

### A. Technical structure

As stated in the introduction, the hardware setup of the proposed system is similar to a usual automation system. Figure 1 depicts the hardware scheme of the approach containing the following components:

1) The overall system requires a *backend* connected to the particular automation network. Moreover, a connection to the *engineering tool* is necessary.
2) One of the main components of this approach is a mobile application (app) running on a mobile device (i.e., tablet or smartphone). This device needs to be equipped with a camera and a touch screen. Furthermore, it is required that the mobile application is connected to a *backend* via IP (e.g., WiFi).
3) The central components are the automation devices. These devices need to be either able to emit visual light (e.g., LED indicators of a light switch) or they control dislocated lights (actuator controlling particular lights in a room). The most important requirement is that the lights are controllable by the automation system.

### B. Identification process

Based on the previously introduced hardware setup, the identification process follows the following procedure:

1) Every automation component is integrated into the automation system off-the-shelf. The backend assigns each component a temporary unique address.

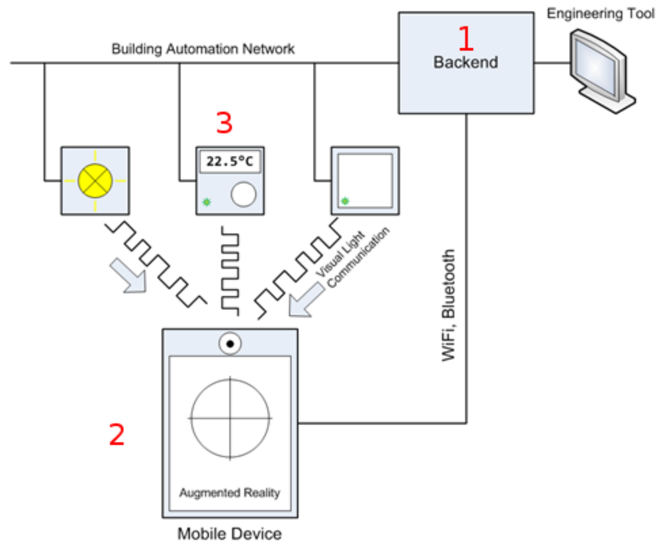


Fig. 1. Schematic view of the solution

2) For each component and according address a locally unique signature is defined by the backend. This signature is transmitted as an optical code, integrating redundancy and failure correction mechanisms. If a single device consists of more parts which need to be identified (e.g., an actuator with 8 ports), the device will receive multiple different codes.
3) By use of the app, the configuration mode is activated.
4) The backend requests the relevant components to emit their individual code(s). The rate of this emission is defined by the technology of lighting in use at the device as fast signal emitting devices (e.g., LEDs) can issue the signal faster than others (e.g., neon tubes).
5) By using the mobile device's cam in combination with electronic image processing software, (sub-)components can be identified based on the detected codes and associated with the temporary address.
6) Once a code is detected (as shown in Figure 2), the user is informed by issuing a message on the display of the mobile device. The selection of a recognized device within the cameras' view starts the provisioning of the component.

### C. Light emission coding

*1) Line codes:* To transmit the payload (i.e., the unique signature generated by the backend), (digital) line codes have to be used. The following criteria (defined in [5]) must be met by the selected line code:

1) **Elimination of DC component**: The DC component causes problems, e.g. on circuits with capacitive coupling or transformers in it. Since communication via light is used, there is no need to eliminate the DC component.
2) **Transmission bandwidth**: This criteria is important for the use case. Depending on the available emitter, the


Fig. 2. Smartphone app, that has just identified a component by detecting its light signal emission

bandwidth may be very limited. In addition the camera on the mobile device may impose some limitations. On the other hand it is desired to transmit as fast as possible in order to get reasonable recognition times (i.e., the time for which the user has to focus the camera on a single emitter).

3) **Transmitter power/Power efficiency**: During the lifetime of the system, only in a short period of time during engineering the proposed communication technique is used. In addition, almost all components which emit optical codes are line-powered. Although there may be some battery powered components, it is assumed that energy efficient light emitters are used (e.g., LEDs). It is therefore not necessary to select a line code which is optimized with respect to transmitter power.
4) **Error detection/correction**: Error detection is necessary to avoid a false recognition. Like in traditional engineering workflows it would introduce a lot of additional effort if a wrong configuration is made.
5) **Timing extraction**: Since there is no additional channel for transmitting the clock or any other external clock source available, the line code must carry timing information.
6) **Transparency**: If certain words/bit combinations are not allowed by a line code, the backend can avoid assigning such words as unique signature to the components. However, it would decrease system complexity if the line code is fully transparent.

Table I summarizes the required line code criteria.

The approach presented in this paper communicates via visual light emitted by some source on the device which has to be identified. External disturbances such as the sunlight or other light sources are very likely and prevent the reliable recognition of different brightness levels. In addition, not every potential light source (e.g., a neon tube) is able to glow in different (defined) brightness levels. Therefore, only two states

TABLE I
LINE CODE REQUIREMENTS

| Criterion | required |
|---|---|
| DC free | -- |
| Bandwidth | + |
| Power | -- |
| Error detection | ++ |
| Timing extraction | ++ |
| Transparency | + |

of the light emitter can be used for communication: dark and bright representing 0 and 1.

Thus, line codes using 3 or more levels for data transmission (like BRZ or 4B3T) can be ignored for further investigation. The few remaining widely used codes NRZ-Code (non return to zero), unipolar RZ-code (return to zero) and Manchester Code (also known as dipolar or split phase), will be compared in detail.

*a) NRZ-Code:* In this technique a 0 is mapped to low, while a 1 is mapped to high. An advantage is its simplicity and its (in contrast to later solutions) better transmission bandwidth since the full frequency can be used. However, since long periods of the same level are possible, techniques like bit stuffing (i.e., after a certain amount of a constant value an inverse element needs to be inserted by the sender; detected and removed by the receiver respectively) are required to remain clock synchronization. Furthermore, timing extraction is difficult.

*b) Unipolar RZ-Code:* Normally, RZ-Code works such that the signal will always return to zero after half a period. Therefore three levels ($-1$, 0 and 1) are required. Using unipolar RZ-Code only two levels (low and high) are needed. A logical 1 leads to a peak to high in the first half period and a return back to zero in the second half. For a logical 0, the signal level does not change. A disadvantage – therefore – is that a long period of zeros again leads to a loss of synchronization requiring additional techniques like bit stuffing. Another problem is that the transmission bandwidth is reduced to half of the possible frequency.

*c) Manchester Code:* In this coding, every bit causes a state change. When transmitting a 1 the line will be high for a half period and low for the rest of the period. Transmitting a 0 will lead to the inverse pattern.[1] The advantage of this coding is that synchronization is always given (i.e., timing can be extracted from the code). However, like unipolar RZ-Code, the bit rate is only half the possible frequency.

*2) Data Representation:* There are many options how to represent the data, including how to detect and correct errors, being briefly described next [6]:

*a) Repetition codes:* Visual emission of the device's assigned code is continuously repeated while in configuration mode.

[1] Depending on the agreed definition, the interpretation of 0 and 1 might be inverted.

*b) Error detection and correction:* Single parity bits are an easy form of checksums calculated out of the data. At the beginning, somewhere between or after the data, those parity bits are inserted which signal that a predefined data block has odd or even parity (i.e., odd or even number of ones/zeros). When using Cyclic redundancy checks (CRC), every block gets a short check value attached, which is calculated by the sender. The check value is typically the remainder of a polynomial division which is applied to the data block. The receiver does the same operation on the data to get the CRC value of the received data and compares the calculated and the received CRC value. Depending on the requirements (i.e. how many errors should be recognizable and correctable) the length of the CRC value vary.

### D. Configuration types

Depending on the automation system and the associated engineering procedures vary. Basically, the approach presented in this paper is usable in several project setups.

*1) Pre-Engineered Configuration:* If all automation devices (and maybe even the functionality) of the facility are pre-engineered and available in an existing configuration (e.g., an ETS project for KNX), only the *identification step* (as described in Section II-A) of the installed components is missing. When the mobile device identifies an automation component using the emitted light code, the information from the existing configuration can be used to present meaningful information and names to the user. For example, if the mobile device recognizes the code of a 2-gang push-button, it shows the names of all engineered but not yet identified buttons of the same type. The user can select one of the presented buttons and the *identification* for this component is finished.

*2) On-Site Configuration:* If the automation devices have not been engineered in advance, the approach can be used to acquire data directly from the field. The gathered information can be imported in an engineering tool in order to complete the engineering. When the mobile device identifies a automation component using the emitted light code, the user can describe the identified component by some meaningful name or additional information. After the transfer of the data to the engineering tool, *binding* and *parameterization* (as described in Section II-A) can be done in the engineering tool.

*3) Ad-Hoc Binding:* In addition to III-D2, an easy binding for predefined functions can be done directly on the mobile device. Based on predefined templates (e.g., lamp ↔ button or fan-coil ↔ control panel), binding takes place by directed identification.

To that end, the user chooses a template, identifies the related components using their emitted light codes and the function defined by the template will be assigned immediately by the backend. This process includes the engineering tasks *binding* and parts of *parameterization* as described in Section II-A.

## IV. Proof Of Concept

In order to verify the feasibility of the presented approach, a proof of concept was implemented based on a Java-based backend implementation and a native mobile application. As automation system KNX was chosen as it allows easy interoperation and configuration.

### A. Backend implementation and issues

As backend an OpenMUC [7] application was developed. OpenMUC is a Java framework based on OSGi that allows controlling different kind of bus systems via the same interface [8]. As frontend an application for Android running on a Samsung Galaxy Tab S for the optical emission detection was implemented.

For the optical signal emission of the different KNX devices, a Manchester encoded signal is used for transmitting the device addresses and a parity bit for error detection is used. The decision for this code was made due to self synchronization capabilities and the easy programming (e.g., there is no need for bit stuffing). For the PoC, where only a 4 bit identifier is being used, a single parity bit is sufficient. However in real use cases where far longer identifiers are needed, other error detecting techniques are advisable.

The process described in section III-B does not need a jitter- and delay-less communication between the server and components. Since this approach is not yet integrated natively into KNX devices, it was necessary in the PoC to cyclically send on/off command to the different devices in order to generate the light emitted code. This imposes limits regarding the flashing frequency of the LEDs: Due to the limited sending frequency of the KNX bus, it was only possible to reach a maximum frequency of 2 Hz, being the sole frequency used throughout this PoC as only LED are used.

### B. Frontend implementation and issues

To identify the different KNX devices we used OpenCV 3.1 [9], a computer vision library for Android and Windows, to detect their transmitting code on the smart phone/tablet.

One of the main difficulties was the fine tuning of the detection since it is based on detecting brightness. Therefore, the HSV (Hue/Saturation/Value) color spectrum was used instead of the RGB spectrum. RGB states how many parts of red, green and blue are contained. To detect different kinds of LEDs (i.e., differently colored LEDs) we only want to care about the brightness (represented by the ”value” in HSV), which is far more easy using the HSV color spectrum.

If the LEDs of multiple devices or channels are within the vision of the camera, all of them will be detected at the same time. At any time regions which are brighter and larger than a threshold are detected. After the detection of a bright area it is reduced to its center-point which is stored. If previously such an area had been detected around this center-point (within a small threshold), it is assumed that the same LED was detected and thus its (new) position will be the updated center point. If an LED is not detected for a short period of time, the position remains stored in order to recognize the off-phase of the signal. If the LED has not been detected for a longer period of time, it is removed from the list of detected LEDs.

We had to experiment with many different thresholds and the camera resolution to find a good compromise and accurate detection of the emitted code. As described, unfortunately we were only able to transmit the code with 2 Hz. Without that restriction, a compromise on the frequency needs to be found as low frequencies tend to annoy the user when using long codes; high frequencies on the other hand may not be possible to detect. The described issues most probably depend on the camera of the mobile device in use and – therefore – need to be further considered after the PoC.

### C. Proof of Concept function

In the PoC, the initialization is started by switching on the recognition mode on the tablet. This command is issued to the OpenMUC framework, that starts to send the respective on/off commands periodically to all devices.

Once an identifier is detected, the application makes a screenshot and sends the detected identifier to the server, which replies with additional information about that particular device. Since more than one device can be recognized at the same time, the screenshot helps the user to identify the different detected devices. Depending on the available information the user can name the device (sensor or actor) and connect it with other devices.

Rules are defined for device types, e.g., such that it is possible to connect a button with a lamp or a dimmer with a lamp, but not to connect a lamp with another lamp. The connections will be stored server-sided but can be reset by the application. Successful configuration is signalized to the user by using optical feedback.

## V. Conclusion

The presented approach introduced a new provisioning scheme for building automation systems. It facilitates existing and well accepted base technologies such as mobile devices and does not require hardware modifications of automation devices. This approach simplifies the identification process of automation components and can be integrated in different types of engineering processes. Compared to the current state of technology it allows for saving time during the engineering process. Due to the immediate optical feedback and the intuitive identification of data points by augmented reality reduces the probability of erroneous configurations.

Although the proof-of-concept showed the feasibility of the approach, automation technologies and standards need to be extended in order to allow productive use. Existing engineering tools need to be extended for being able to make use of the concept. When integrating this concept in a productive system, security aspects have to be considered. Due to the use of mobile devices and an additional communication channel, the attack surface is increased (cf., [10]).

Based on the experiences learned during the proof of concept, time savings during the installation can reach up to two third. Of the four engineering tasks introduced in Section II-A, this savings are the highest during the identification phase. In a future step also frequency, efficiency and redundancy of the visual signal have to be investigated. This strongly influences the usability and robustness of the system, but strongly depends on the physical properties of the lighting technology. In order to increase the usability and the degree of automation, e.g. localization systems can be used [11], [12]. This can support the identification and functional mapping.

## Acknowledgments

The presented work is based on a pending patent and is conducted in the "OptiMAS" project, funded and supported by the Austrian Research Promotion Agency (FFG).

## References

[1] J. Hendrix, N. Mcfarland, G. Nass, J. Raimo, and P. Songkakul, "Binding methods and devices in a building automation system," Mar. 6 2008. WO Patent App. PCT/US2007/077,107.

[2] T. Komine and M. Nakagawa, "Fundamental analysis for visible-light communication system using led lights," *IEEE Transactions on Consumer Electronics*, vol. 50, pp. 100–107, Feb 2004.

[3] W. Kastner, G. Neugschwandtner, S. Soucek, and H. Newmann, "Communication Systems for Building Automation and Control," *Proceedings of the IEEE*, vol. 93, pp. 1178–1203, Jun 2005.

[4] W. Kastner, L. Krammer, and A. Fernbach, "State of the Art in Smart Homes and Buildings," in *Industrial Communication Technology Handbook, Second Edition* (R. Zurawski, ed.), ch. 55, CRC Press, Inc., 2014.

[5] I. Glover and P. Grant, *Digital Communications (2nd Edition)*. Pearson Education Canada, 2003.

[6] A. A. Bruen and M. A. Forcinito, *Cryptography, Information Theory and Error-Correction. A Handbook for the 21st Century*. John Wiley & Sons, Inc., 2005.

[7] Fraunhofer Institute for Solar Energy Systems, "OpenMUC website," 2016. http://www.openmuc.org/.

[8] S. Feuerhahn, M. Zillgith, R. Becker, and C. Wittwer, "Implementation of an open smart metering reference platform - OpenMUC," *ETG-Kongress*, 2009.

[9] Intel, Willow Garage, "OpenCV website," 2016. http://www.opencv.org/.

[10] W. Granzer, F. Praus, and W. Kastner, "Security in building automation systems," *IEEE Transactions on Industrial Electronics*, vol. 57, pp. 3622–3630, Nov 2010.

[11] E. Martin, O. Vinyals, G. Friedland, and R. Bajcsy, "Precise indoor localization using smart phones," in *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, (New York, NY, USA), pp. 787–790, ACM, 2010.

[12] A. K. M. M. Hossain, H. N. Van, Y. Jin, and W. S. Soh, "Indoor localization using multiple wireless technologies," in *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems*, pp. 1–8, Oct 2007.