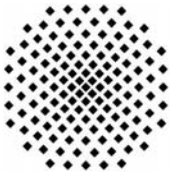

Universität Stuttgart

Fakultät Informatik, Elektrotechnik und Informationstechnik

Processes for Human Integration in Automated Cloud Application Management

David Schumm[1], Christoph Fehling[1], Dimka Karastoyanova[1], Frank Leymann[1], Jochen Rütschlin[2],

Report 2012/02
February 21, 2012

[1] Institute of Architecture of Application Systems
Universitätsstraße 38
70569 Stuttgart
Germany

[2] Daimler AG
Epplestraße 225
70546 Stuttgart
Germany

CR: C.0, C.2.4, D.2.2, D.2.3, D.2.7

General Remark

This work is based on research conducted at the Institute of Architecture of Application Systems (IAAS) in the field of human integration in e-science applications and scientific workflows, as well as on collaborative work and discussions with Daimler employees. The results and recommendations given in this report are based on this collaboration. Best research practices have been applied to draw the conclusions presented in this report, but nevertheless, the applicability of the results and recommendations must be verified in each concrete context.

Index

1 Introduction 4
2 Human Communication Flows 6
2.1 Human Communication Flow Essentials 6
2.2 User Response Required 11
2.3 User Notification 13
2.4 User Response Optional 15
3 References 17

1 Introduction

Cloud computing has introduced elastic runtime infrastructures, in which IT resources, such as servers, may be reserved and released within minutes. These commodity resources are provided to very large customer groups allowing cloud providers and customers to equally benefit from economies of scale: even though cloud resources are provided using a static IT infrastructure, the number of users enables the cloud provider to offer pay-per-use pricing models. Therefore, customers only pay for the actually used resources and can adjust the number of resources very quickly to respond to the workload currently experienced by an application hosted in the cloud. In this scope, beneficial effects are maximized if resource management is automated so that the application itself may independently determine the number of required resources at any time of its execution.

Some decisions, however, may still require human interaction. In the context of potentially self-managed cloud applications, these human decision makers should be efficiently integrated. To ensure the beneficial effects of a cloud environment, this integration has to be realized using modern communication channels in order to ensure minimal response times for human decisions. Otherwise, the flexibility and dynamicity of a cloud infrastructure is significantly hindered. As a consequence, we propose in this report to explicitly model human activities in automated management flows invoked by the self-managed cloud application.

Figure 1 shows the abstract cloud management architecture that we assume in this report. In the following we provide a description of the fundamental elements of this architecture, before we present the centerpiece of this work: the human communication flows which provide for seamless and efficient integration of humans into applications.

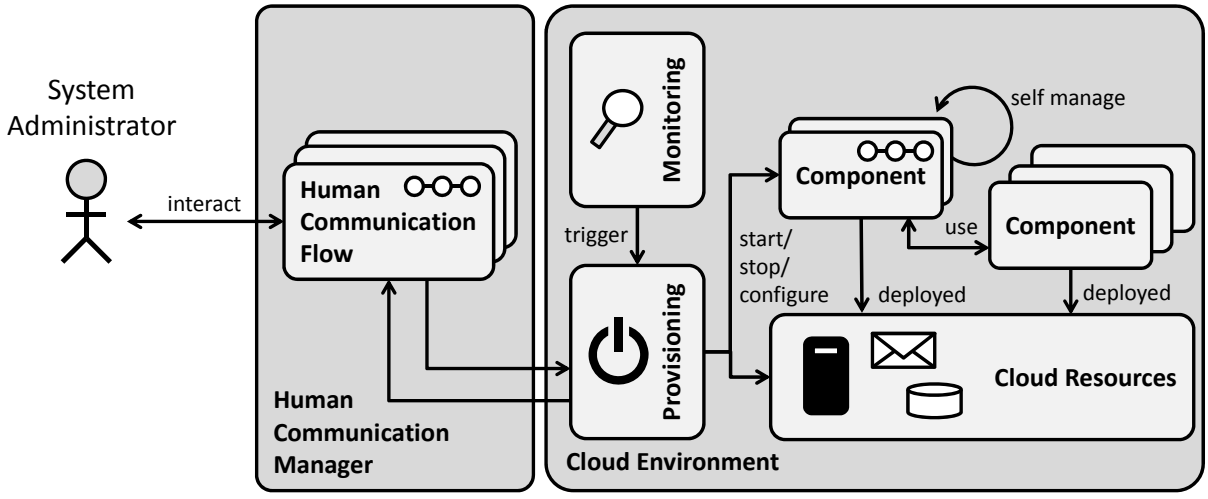

Figure 1: Abstract management architecture.

Components

We consider cloud applications to be *componentized*, thus, being comprised of *loosely coupled components* [2]. Application components use each other and are deployed on cloud resources. Management may either be controlled from the outside or from the components themselves in form of self-management.

Cloud Resources

Cloud resources are offered in an *elastic infrastructure* [2]. Any of the cloud types describe in [2] may be used in this scope, which states the cloud resources may reside in isolated *public*, *private* or *community clouds*. On the other hand, they may also be hosted in a *hybrid cloud* environment comprised of different cloud types.

Monitoring

The cloud resources and the cloud application provide runtime information to a monitoring component. This may be realized using two information styles: (i) events may be *pushed* to the monitoring component when certain conditions occur, and (ii) the monitoring component may proactively *poll* information from the managed application and cloud environments.

Provisioning

The provisioning component provides functionality to provision and deprovision cloud resources and application components. To achieve this, it may interpret dependency models, such as [3] [4], or may contain a number or automated scripts or provisioning flows to setup and configure cloud resources and application components in a particular order.

Human Communication Flows

Cloud system management can hardly be automated completely. In many cases, human interaction is required, for example, to approve the execution of certain management flows.

2 Human Communication Flows

In this section, we present general flows and their inherent properties to support communication of cloud-based applications and services with humans. At first, we elaborate on the essential aspects of establishing communication with a human. Further, we discuss general scenarios in which communication of applications with humans is applicable and useful. Then, the communication flows are presented in detail. A communication flow formalizes the different interactions of components which are required to establish communication between an initiator (e.g. a faulting application component) and a human user (e.g. a system administrator).

2.1 Human Communication Flow Essentials

The communication flows described in this section follow a particular terminology, explained in the following. The **participants of communication**, as depicted in Figure 2, are:

1. A *communication initiator*, (e.g. a cloud-based application, a service etc.) wants to establish communication with a human user. Communication is initiated through a communication request message sent to the human communication manager.
2. The *human communication manager,* this component coordinates human interactions in order to establish a communication between a communication initiator and a communication device that is operated by a human user.
3. *Communication services* enable communication between the human communication manager and the device of a user. They support the protocol required by the device.
4. A *communication device* of a user represents a concrete endpoint for a selected communication channel. For example, an E-Mail/SMTP communication channel may use the address “admin@cloudcomputingpatterns.org” as an E-Mail endpoint for submitting requests and receiving responses in form of E-Mail messages.

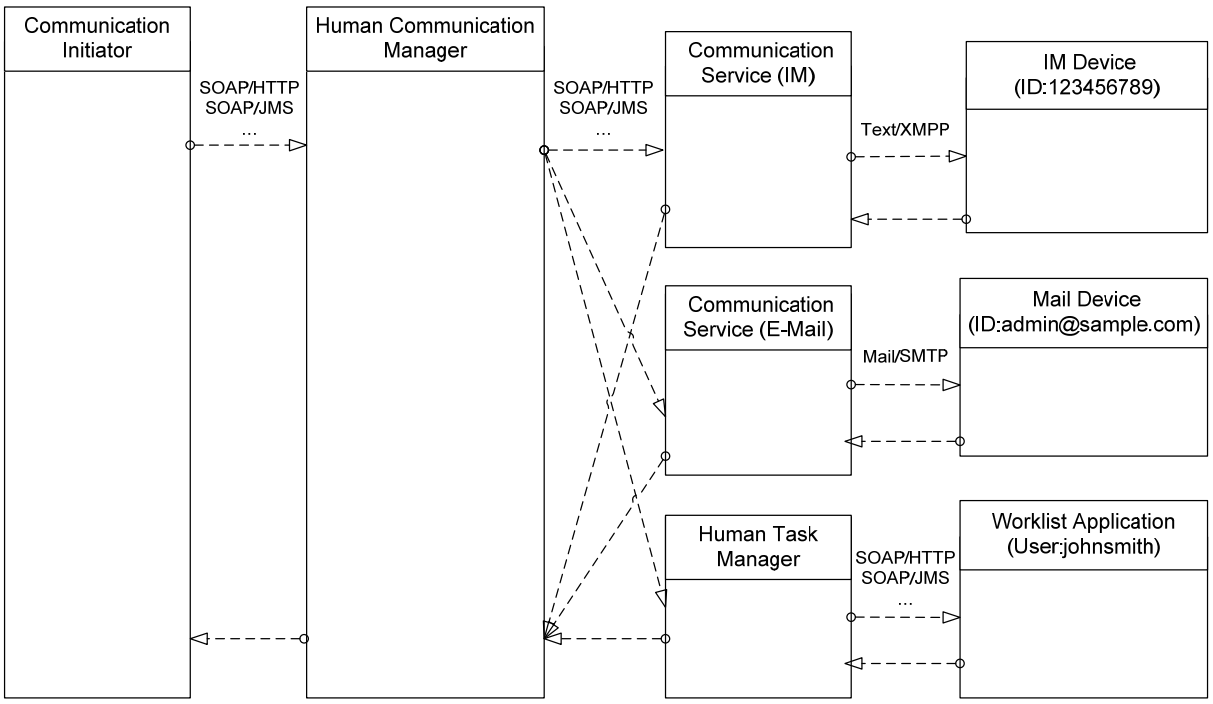

Figure 2: Communication participants.

Regarding the message exchanges between the participants, we can distinguish three major types: Communication request message, communication message, and channel-specific message. A communication request message contains communication parameters which are interpreted by the human communication manager. These parameters indicate the type of communication, details on user and communication channel selection, as well as format and further details about an expected reply. The request also contains the communication message to be delivered to the human user. Depending on the chosen communication channel, the properties of the communication message are rendered differently, for example into the properties of an E-Mail, as depicted in Figure 3.

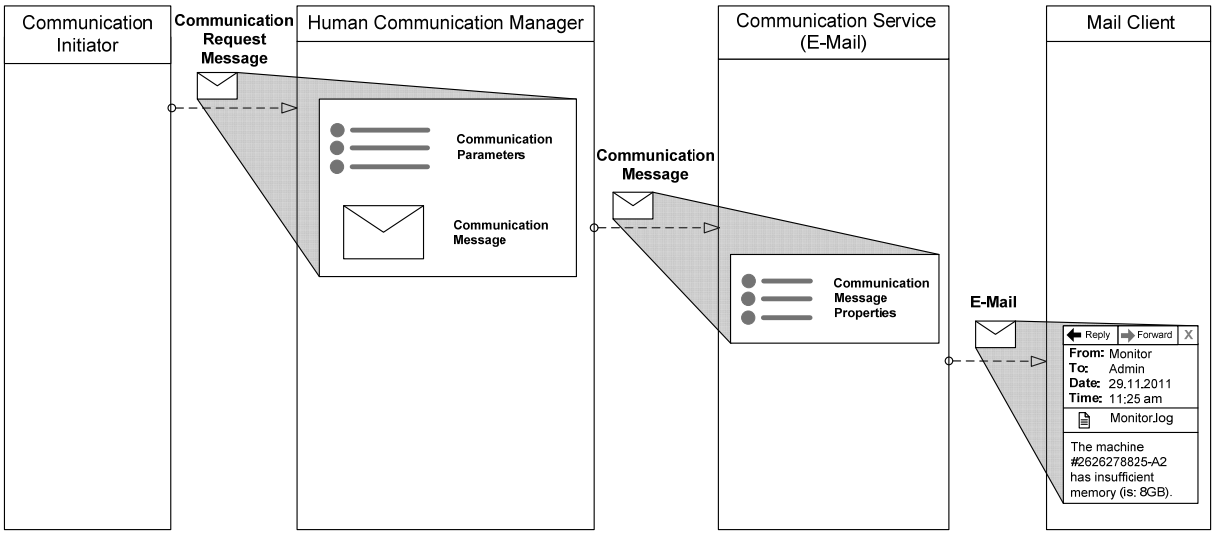

Figure 3: Messages in human communication.

In Figure 2 and Figure 3, all participants of the choreography are shown. To reduce complexity of the communication flows we abstract from the used communication services which connect the human communication manager to the endpoints in the rest of the section.

As proposed in [6], we can distinguish at least three main **classes of communication** with humans: Notification, decision, and control. Communication can be further distinguished through different approaches for user and device selection and parameters like response optionality, communication expiry etc. The parameters are described in detail later in the section.

1. *Notification*: This class of communication refers to reports sent to a user in order to inform him about a particular status of an application, service, workflow etc. The reports may be regular status reports, fault reports, processing completeness updates (e.g. status of a provisioning flow) etc. A notification does not require any response by the user.
2. *Decision*: This class of communication refers to sending information to a user to let him decide how to react to a particular situation. A decision may be a simple approval/rejection, may offer multiple choices to select from, may require specifying particular parameters, may require a user to check or correct certain data that cannot be checked or corrected automatically, or may request, how to handle a particular fault that was thrown within some component of an application (related to class 3).
3. *Control*: this class of communication refers to remote control of an application that can be steered by a user. Depending on the capabilities of an application for remote control, processing may be paused, resumed, aborted, retried in case of a fault, iterated, or skipped.

The communication flows presented in this section focus on notification and decision. The design of flows for control of an application as well as auxiliary flows like registration and deregistration of users and devices, is ongoing research and requires further practical evaluation in human communication scenarios. Also, flows for communication between humans like task delegation are not covered here. In order not to make the flows too complex, data flow is not explicitly modeled. Also, communication cancellation and advanced settings like retries of communication attempts are not included in the flows.

Human communication flows are applicable in various **usage scenarios**: For instance, in the area of scientific simulations, from which the communication classes are originated, there is a strong need for flexible integration of different devices and communication methods in order to integrate humans in automated processing. For example, a simulation run may be started late on Friday to use calculating power available only during the weekend. However, if the simulation aborts with a fault after some hours, one week of time is lost to run the simulation again. In order not to require a scientist to be in office all over the weekend, different ways of communication are interesting. For instance, E-Mails can typically be checked from everywhere. A notification sent to the user this way is a comfortable way to account for such circumstances.

For **usage in cloud-based applications**, this applies as well. The management systems of different cloud-based applications are often scattered across different platforms, requiring various monitors to be opened all the time. Notifications that are sent to one central mail address registered in the human communication manager – the central component in charge of establishing communication with humans – solves this issues. The communication flows described in this section assume that all information required for the interaction with a user is sent to the human communication manager. The human communication manager manages all human interactions and enacts communication flows to route messages to them. We envision communication flows to be based on task classification and presence information [1]. A task classification (e.g. critical fault) may be used to select a set of users which are registered for particular topics, like faults of particular applications. Then, based on presence information about these users (e.g. public holiday, status in Skype etc.) the most feasible communication device that is registered for the user is chosen for communication. For example, if a fault is critical, then the user is immediately contacted via a Skype channel if the user is online. If the user is offline, an E-Mail with high importance is sent.

As mentioned before, we can distinguish different classes of communication: Notification, decision and control. However, communication can be further distinguished by several parameters. In the following list, we describe the **communication parameters** with a user:

1. Type of human communication
    1. Notification: This type realizes the first communication class. It implies that no response from a user is expected.
    2. Response required: This type realizes the second communication class. It implies that a response by a user is mandatory.
    3. Response optional: This type realizes a special case in the second communication class. It implies that a user may respond, but a default behavior is pre-defined that is sent in case the user does not respond.

2. User selection
    1. Pre-defined by the communication initiator: User selection may be pre-defined by the application that invokes the communication manager. Depending on the implementation of the communication manager, this may be a staff query or static user names.
    2. Defined by the communication manager:
        1. Single user: A single user is selected to interact with.
        2. User group: All users contained in a particular user group are selected.
        3. User set: A set of users, possibly from different groups, is selected.
        4. User broadcast: All users of a particular domain are selected.
        5. Presence-based: Users are selected based on their presence information.
3. Communication channel selection
    1. Pre-defined by the communication initiator: The selection of communication channels such as E-Mail, Skype, SMS, File, worklist application etc. can be pre-defined by the initiator of the communication.
    2. Defined by the communication manager:
        1. Single channel: For a selected user, only one channel is used at a time.
        2. N out of m: For a selected user, multiple channels are used simultaneously, for instance to deliver a very urgent message as fast a possible.
        3. Channel broadcast: For a selected user, all known communication channels are used simultaneously.
        4. Presence-based: For a selected user, the channel is chosen that can deliver the message to the user the earliest.
4. Response options
    1. Data type: This mandatory parameter specifies the format of the response expected from the user. For instance, Integer [1..5], Boolean, String [Yes, No], a particular XML Schema etc. are applicable formats. The communication manager tries to parse the response by the user into that format. If this validation attempt is not successful, the user is re-contacted by the communication manager in order to correct the prior response.
    2. Attachments: This optional parameter specifies attachments which are expected from the user. The usage of attachments limits the selection of channels. For instance, an Excel Sheet with parameters can be supplied via E-Mail, Skype, MMS, or a Web form, but not via SMS. The same applies to large attachments.
    3. Communication expiry: This optional parameter defines the deadline, after which a communication expires. After that time, a default response is returned. Responses or corrections by the user arriving after that time are refused with a note about communication expiry.
    4. Default response: This parameter specifies the response that is sent to the communication initiator after communication expiry. This parameter is only required, when a communication expiry is set.

These parameters need to be passed to the human communication manager by the communication initiator. The parameters can either be included in the message headers or in the body of the message that is sent to the human communication manager. These parameters are interpreted in the different activities which are performed in the communication flows.

Besides the communication parameters required for establishing a communication with a human user, the different **communication message properties** are essential:

1. Subject: The subject of the message to be delivered to the user(s).
2. Body: This property represents the message contents to be sent.
3. Attachments: This complex property represents a set of attachments to be transferred to the user. Depending on the communication channel that is selected, the attachments are differently delivered. The form of delivery also depends on the size of the attachments, for instance, a file of 50 Mb is most likely to be provided via ftp or http. Thus, in that case the file would need to be stored on a server and just a link would be sent to the user.
4. Importance: This property can be used for advanced selection mechanisms of users and devices.
5. Date / time: This property represents the date and time the message has initially been issued.
6. Message classification: This extensible property describes the semantics of the message to be delivered to a user. Aside from the basic classes like notification or decision, domain-specific classes can be defined. The semantics of a message can then be precisely classified through multiple characteristics, of which some will likely not be set by the invoking component, but by the communication manager. Examples for domain-specific classes are response urgently required, failure notice (blocking, non-blocking, thread-blocking, path blocking, process blocking, response validation failure), confirmation (response received, response validated, attachment received, attachment validated) request for re-response, registration status (user registered, user unregistered, device registered, device unregistered).

The communication message properties may be rendered differently, depending on the selected communication device of the user. For example, the message subject property is rendered in a different way in an E-Mail, than in a Skype conversation. To name another example, request for acknowledgement can be rendered as read receipt request in an E-Mail, while in a Skype conversation the user needs to type a reply. Furthermore, the message body to be delivered to the human user may be extended depending on the rendering. For example, if a complex data structure (Integer [1..5], Double, Double) is expected, a description of the format to be returned could be appended, including an example.

2.2 User Response Required

How can a system administrator be informed about critical conditions and events of cloud-based applications, when human actions have to be performed in a certain time-frame?

Context
A cloud application that is managed automatically will still run into conditions that require human management activities.

Problem
Some management activities are very hard to automate, so every cloud application will rely on human actions during its runtime. For example, this may involve all physical adaptations of the cloud application's infrastructure, such as replacement of hard drives of resources in a private cloud. Often, these management tasks have to be handled in certain time period. Therefore, the management flow should interact with human administrators and escalate the request if actions are not performed in a timely manner.

Solution
Send information to human administrators and have them acknowledge its receipt and the completion of required management activities. If these acknowledgements are not received, escalate the request by using different communication channels or sending it to more/different human actors.

Result
Timely reactions to critical conditions in the cloud application are ensured by escalating the information stepwise. For example, first, a critical e-mail message may be sent. If no response to this message is received after a certain time interval, other communication channels, such as cell-phones or VoIP systems may be used to establish a more direct interaction with administrators. Should this still be unsuccessful then the request may be escalated to a higher-ranked system administrator and so on.

If this communication interaction flow is integrated into another management flow, interaction should be handled in an asynchronous fashion to ensure that the overall management flow is not blocked while waiting for human interaction. Another critical design issue is determining the request escalation procedure to ensure an adequate behavior of the cloud application with respect to the criticalness of an experienced condition.

Relations to other patterns
If the user shall only be informed about some activity / state in the managed application and no action is required, the "user *notification"* communication pattern may be applied.

Variations
The contact user activity may try to reach the user via different devices one after another.

Known uses
Simplified variants of this flow are common in systems management systems to inform system administrators that their immediate action is required.

Sketch

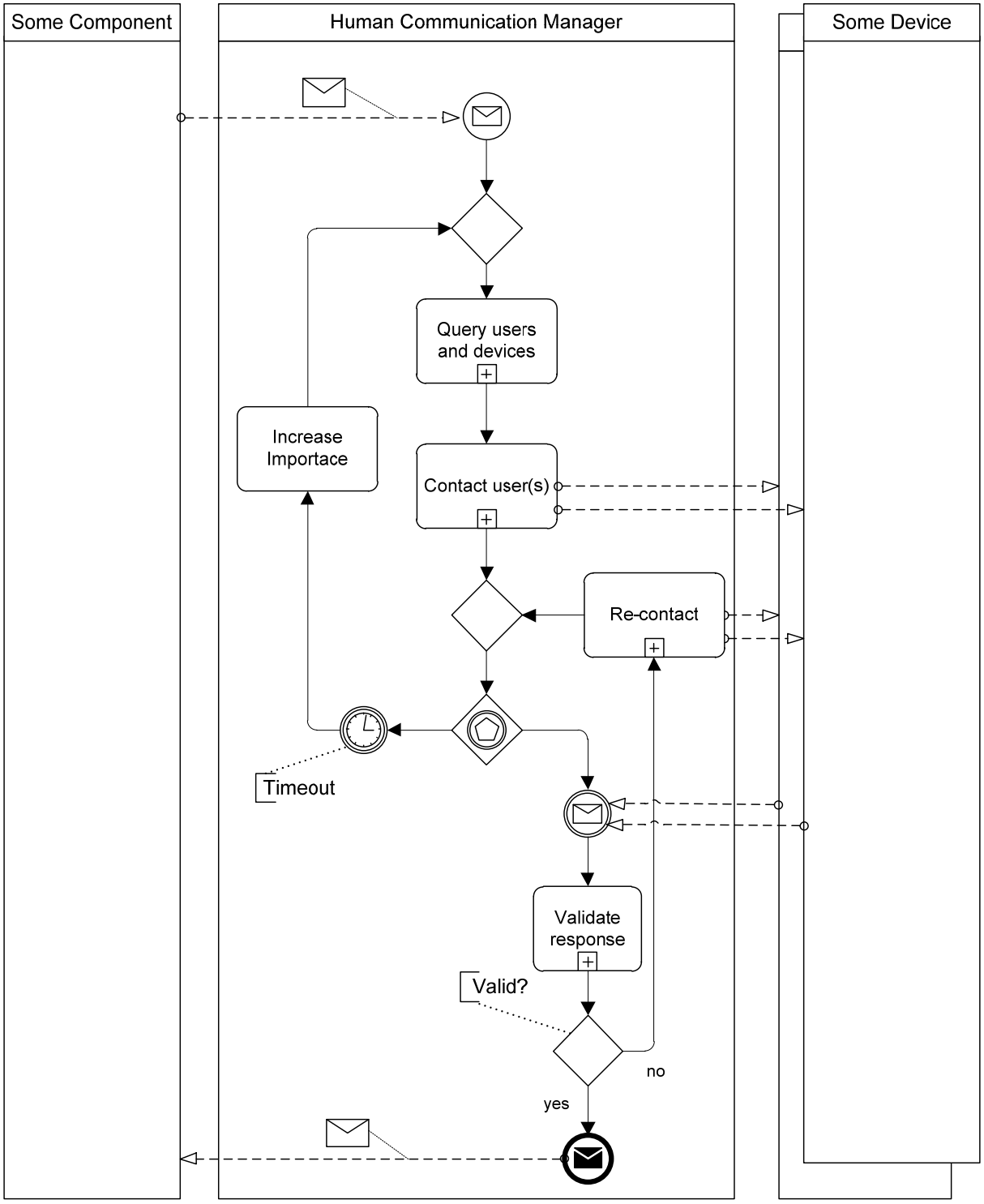

Figure 4: Human communication flow – user response required

2.3 User Notification

How can a system administrator be informed about certain conditions and events of a cloud-based application?

Context

A cloud application whose management has been partially automated needs to inform a human administrator about certain conditions or state changes.

Problem

Even if most of the management tasks arising for a cloud-based application have been automated, certain conditions will always require the interaction with human administrators. This interaction can be purely informational, but monitoring information can also be used to audit the system performance and to identify future problems and bottlenecks.

Solution

Include activities in management processes that inform human system administrators about the state of the cloud-based application as well as the management tasks that have been handled automatically.

Sketch

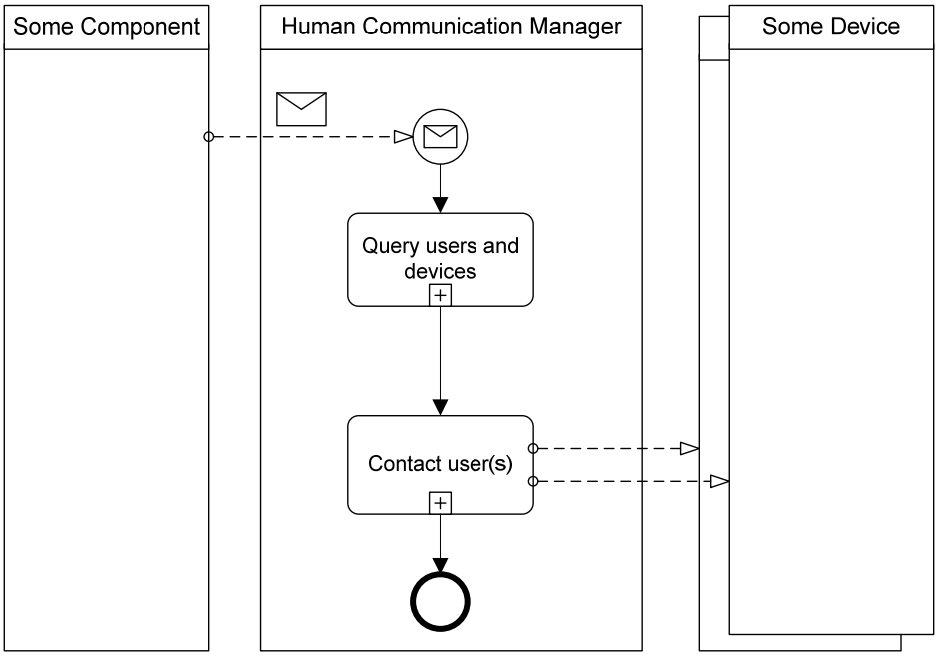

Figure 5: Human communication flow – user notification.

Result

Human system administrators are now informed about important conditions and events in the system and may analyze its performance. In this scope, different classifications of informational messages should be used, for example, informational, warning, error etc. to allow a better filtering, priority sorting, and routing to particular experts. In the case of required concrete actions from the user the management processes must interact with the administrator to ensure that the required actions have been performed. The flows for user response required (Section 5.3) and user response optional (Section 5.4) handle such management tasks.

Relations to other patterns

The "*user notification"* communication pattern may be combined with any other management flow. In this form, it is initiated from any of the activities of the management flow it is combined with to inform the user that the corresponding activity has been started, stopped, has failed etc.

Variations

Instead of selecting the device from the human communication manager, the information may also be published through a communication channel that is accessible by different devices, for example, via RSS [5]. The actual device used to access the information, is then determined by the user individually.

Known uses

Functionality supported by the notification flow is commonly implemented by call centers, workflow engines, and customer relationship management systems.

2.4 User Response Optional

How can a system administrator be informed about certain conditions and events of cloud-based applications, when a default system behavior is pre-defined, but possibly not optimal?

Context
A cloud application whose management has been partially automated needs to inform a human administrator about certain conditions and query how to proceed.

Problem
Cloud applications that use automated management flows are variable to a certain degree. When multiple options are available on how to proceed, the problem is to decide what the best option is. In some cases, an experienced system administrator knows an option that is better than the default option.

Solution
Include activities in automated management flows to request a system administrator to choose one of multiple options. After a certain timeout has been reached, the default option is selected and carried out accordingly.

Result
The human communication flow ensures that the administrator has the option to choose optimal system behavior in uncertain situations. If there is no response from the administrator after a certain time, a default selection is returned.

Relations to other patterns
If the user shall only be informed about some activity / state in the managed application, the "user *notification"* management flow may be applied. If a response of the user shall be enforced more strictly, the "*user response required"* management flow can be used.

Sketch

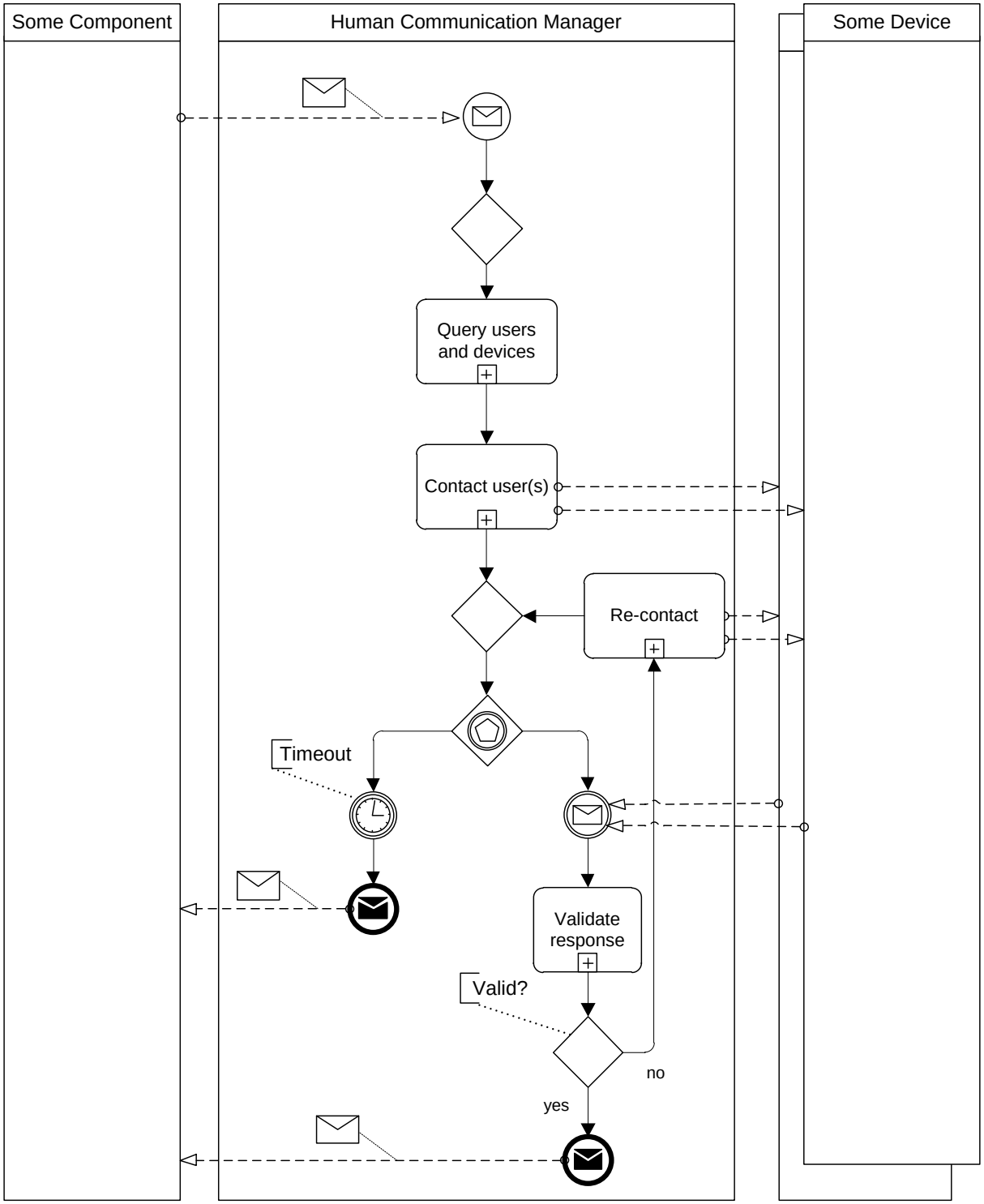

Figure 6: Human communication flow – user response optional.

3 References

[1] Day, M.; Rosenberg, J.; Sugano, H.: A Model for Presence and Instant Messaging, Network Working Group RFC 2778, 2000.

[2] Fehling, C.; Mietzner, R.; Leymann, F.: A Collection of Patterns for Cloud Types, Cloud Service Models, and Cloud-based Application Architectures. Technical Report, 2011.

[3] Mietzner, R.: A Method and Implementation to Define and Provision Variable Composite Applications, and its Usage in Cloud Computing, Ph.D. Thesis, 2010.

[4] Mietzner, R.; Leymann F.: A Self-service Portal for Service-based Applications. Service-Oriented Computing and Applications (SOCA), 2010.

[5] RSS Advisory Board. RSS 2.0 Specification, 2009. Available at: http://www.rssboard.org/rss-specification

[6] Schumm, D.; Karastoyanova, D.: Integrating Humans in Scientific Workflows: Integrate, Register & Communicate. Poster at the 4th Simtech Status Seminar, 2011.