Model-Driven Architecture for Self-Adaptive Context-Aware Message Routing in Pervasive Environments

Nachoua Guizani
SFR Santé Lyon-Est, ERIC Lab
Université Lyon 1, Université Lyon 2
Lyon-Bron, France
e-mail: nachoua.guizani@gmail.com

Jocelyne Fayn
SFR Santé Lyon-Est : eTechSanté
INSERM US7, Université Lyon1
Lyon, France
e-mail: jocelyne.fayn@inserm.fr

***Abstract*—Ensuring suitable message transmission Quality of Service (QoS) still remains among one of the most crucial requirements, especially in case of emergency. A message routing policy application should face the challenge of dynamic source and target context changes, such as resources availability and environmental conditions, and adapt its behavior and decisions accordingly. In this paper, we propose a context-aware model and a functional architecture aimed to steer an intelligent, personalized and adaptive message routing policy. Our objective is to enhance at the operational level message transmission QoS in terms of delivering the right message to the right recipient with the right delay requirements, by taking into account message, sender and recipient ecosystems. In the proposed architecture, we highlight how a message routing policy can reason about context information and adapt autonomously its behavior in response to unpredictable events and context changes in pervasive environments. This architecture is based on ambient intelligence and complies with different scenarios. The relevance of our approach is demonstrated by a use-case in the eHealth domain.**

Keywords-context aware systems; adaptive system; ecosystem; message routing policy; Ambient Intelligence.

I. Introduction

Various applications in the computing field perform multiple tasks, which may require a reliable coordination and communication between different actors. These tasks produce a workflow chain embedding a large number of messages which have to be properly routed and timely handled, especially in emergency scenarios. Several emergency real-life situations are subject to quite high failure rates because of poor communication infrastructures and uncontrolled and non-adaptive message routing policies. Messages are usually blindly transmitted to remote recipients without prior knowledge of their contextual environment availability, reliability and capability. In addition, in some cases the relevance of the recipients should be checked with respect to the urgency and to the message context in order to efficiently and rightly forward it. Several context changes can occur at run-time caused by mobility of users and devices especially in pervasive environments. To overcome these difficulties, we need to set up reliable and adaptable message routing policies ensuring an intelligent message exchange. This policy has to deal proactively with unpredictable events, such as recipient unavailability and exceeding the message treatment deadline, to continuously take into account context changes (e.g., climate, localization, etc.) and to adapt the message delivery accordingly.

The concepts of context-aware [1] and adaptive systems [2] are among the most exciting topics in ubiquitous computing [3] today. To achieve high levels of awareness and adaptivity in message exchanges among two different environments, the challenge is threefold: model source and target contextual information, identify the constraints called adaptation situations to which routing policy applications are sensitive [4], and adapt routing policy behaviors according to context changes.

In this paper, we propose, as a first step, a context-aware routing policy model showing the different objects involved in the message routing processes, as well as the relationships between them. Moreover, we propose a functional architecture of a distributed, context-aware system dealing with message routing policy management in pervasive environments. This architecture is based on AmI ambient intelligence [5] technologies.

Our main objective is to ensure, at operational level, message transmission Quality of Service (QoS). We define QoS as the ability of delivering a given message to the right recipient while satisfying delay and context constraints (availability, experience, trust, etc.). Indeed, we focus on making the routing policy: (*1) Personalized*: in the way that it routes the message to the most relevant destination according to context analysis and, for each message, determines the required delay for reception, reading and reply; *(2) Context-aware*: routing policy application is sensitive to context; (*3) Adaptive*: means that it is able to adapt its behavior in real-time according to context changes; *(4) Intelligent*: in the sense that we take advantage from AI technologies to integrate some intelligence in the routing policy decisions. To summarize, we aim to empower systems to autonomously deliver the right message to the right individual with the right delay requirements, by taking into consideration message, expeditor and recipient contexts.

The paper is further structured as follows. The next Section discusses related work. In Section 3, we present an UML model describing a context-aware message routing policy. In Section 4, we propose a context-aware architecture ensuring intelligent routing policy management. Section 5 shows a case study in the eHealth domain illustrating the message routing policy behaviors in an emergency scenario.

 ISBN: 978-1-61208-405-3

II. RELATED WORK

Several recent research papers spotlight workflow exceptions handling and routing policy management. [6] proposes a proactive detective control model to prevent possible shutdown and violations in workflow applications, and highlights the capacity of Service Level Agreement to ensure/provide QoS avoiding cloud services composition failure and improving the dynamicity of workflow execution. [7] takes advantage of context-aware systems to solve wireless local area network routing problems. However, in this paper the authors restrict the context to device energy-oriented context. The same yields for [8], which proposes an adaptive QoS and energy-aware routing approach for Wireless Sensor Networks (WSNs) based on an improved ant colony algorithm. Several other papers also demonstrate that efficient workflow management under unpredictable events effectively contributes to QoS improvement at different levels and in different fields. A typical example in the eHealth domain can be found in [9]which proposes a framework for modeling context-aware workflow driven resource allocation based on Petri Nets.

Several works have also addressed the context-aware computing paradigm, which becomes an important research issue especially with the emergence of ubiquitous computing [3]. According to [1], context-aware systems are a category of systems that adapt their behavior at run-time according to their users' needs, by proactively anticipating the users' needs without explicit user intervention. To deal with decoupling applications from context information layers, several middleware have been proposed in the computing literature. CAMidO [10] is a Context-Aware Middleware based on Ontology. The particularity of this middleware was to provide a metamodel for context description. The author's idea was to monitor significant context changes and consequently, to dynamically adapt the applications to react to these changes. Both, collection and adaptation are carried out based on an ontology representation. Context-Aware Middleware based on a context-awareness Meta-Model [4] is another middleware and run-time model for dynamic context management based on a model-driven architecture. The paper shows how applications can dynamically adapt their behavior at run-time according to context changes. [11] proposes Unified Context-Aware Application Model, a generalized context-aware architecture for heterogeneous smart environments. The context representation is ontology-based and deals with the 6 types of questions: Who, What, Where, When, Why and How.

Obviously, context-aware approaches have also been adopted in autonomic computing and self-adaptive systems design [2]. Self-adaptive systems aim at ensuring dynamic behavioral adaptation with respect to context changes. Adaptation in the computing literature can operate on four elements: service, interface, content, and software components. [12] proposes a dynamic adaptive service dealing with both highly dynamic changes in pervasive environment and limited resources. Three major steps shall be followed when designing self-adaptive systems: adaptation modeling, analysis and validation. [13] proposes a context Petri Net model for improving the correctness of the configuration of self-adaptive systems aimed at verifying reachability and liveness as key priorities. [14] combines Aspect-Oriented Models and run-time models to design an adaptation model for correct system configuration processing at run-time. The adaptation model includes a set of adaptation rules, which have been introduced to change the system behavior during execution.

Unfortunately, the issue of performing intelligent, adaptive and personalized routing policies has not been adequately treated by the presented research works. Furthermore, the multidimensional aspect of workflow management has not been taken into account. In the architecture we are proposing in this paper, we have taken these issues into consideration and we will build on AmI [5] to ensure a reliable message routing policy aware of the message, sender and receiver contexts.

III. CONTEXT-AWARE ROUTING POLICY MODEL

In this Section, we present our proposed context-aware message routing policy model (see Figure 1), as well as the terminology that we use in this paper. The model points out several concepts involved in message routing processes and the relationship between them.

As shown in Figure 1, a message routing process involves several entities: source actor also called message sender, target actor named receiver, and the message itself. Each entity is surrounded by its own environment and situations under which the message routing policy may change its behavior. A routing policy should provide different techniques and rules necessary to ensure data transportation from their source point to a target point. We distinguish two types of parameters required for message routing.

1) Preprocessing parameters: they should be known before message routing, such as the nature of the required destination, the message routing means (PC, phone, etc.) and type (SMS, mail, etc), etc.

2) Processing parameters: they are determined to control the routing process, such as the required delay for message reception, reading and reply.

The model reflects the multidimensional aspect of the message exchange problem. It highlights the three ecosystems to which the routing policy is sensitive: the source, target and message ecosystems. An *ecosystem* is a set of complex and scalable information systems that are related to entities in a given environment. The routing decision making should be driven by ecosystem data. As defined by [15], *an entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves".* In our case, we consider the entity as a physical or a logical element represented by a person, place or object which is involved in the message routing process. More generally, an *entity* in the message routing process can play several roles (sender, receiver, the concerned parties, author, etc.) in the triggering of an event (order, request, etc.), which generates a message transmission. Each message has its own ecosystem.

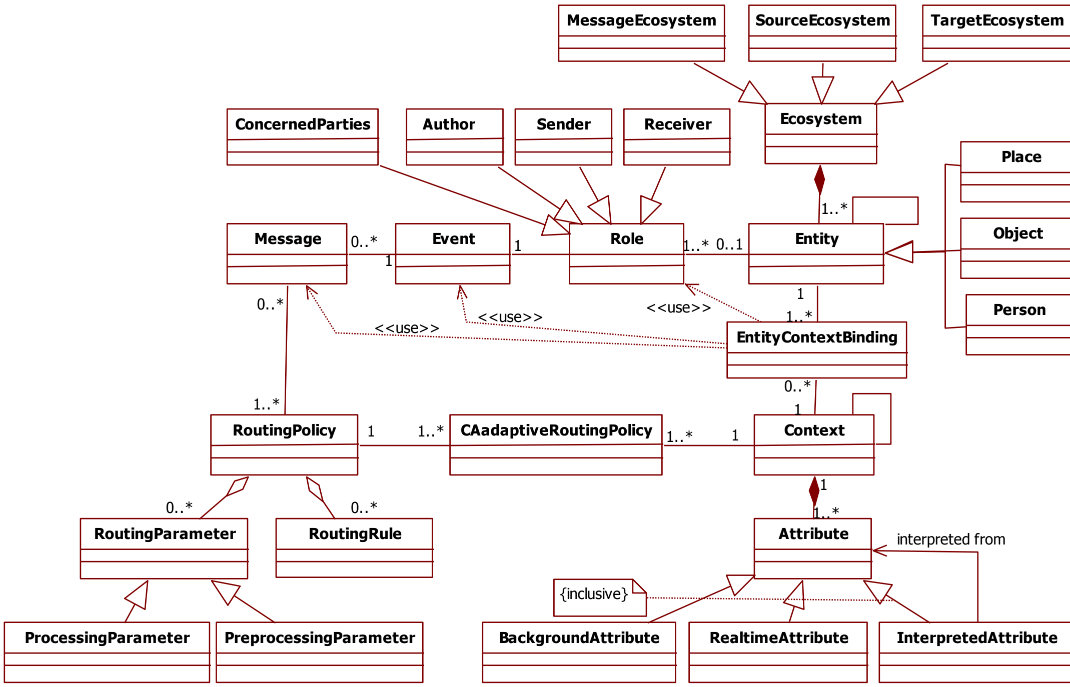

Figure 1. Context-aware message routing policy model

We distinguish different types of messages. Their content could be a piece of information or an alarm. They can be intentionally sent on request of a person, or automatically invoked or triggered by a device in case of a new event or of some failure. An ecosystem is composed of a set of entities.

An *entity* can be related to another. It can be related to a wide variety of type of *contexts* (geographic, climatic, medical, social, etc.) through an automatic entity context binding, knowing that a context can have an impact on another context. An *entity context binding* is like a bridge built at run-time during a context collection process. This binding allows answering to the following question: For a given scenario, which relevant context information do we need to collect? The choice of the context to take into account in a message routing scenario will depend on various factors, such as the content and subject of the message, and the role of the actors in message sending. For instance, in a medical emergency scenario, a bridge needs to be created at run-time between the environmental contexts (hostility, weather, access) of the target and the source, since there is a high probability of persons transfer. Each context corresponds to a multitude of attributes, also called contextual information or observations in context-aware systems. An attribute can be interpreted from a set of basic attributes or not. For instance, climatic conditions depend on different parameters: temperature, snow, etc. In addition, we distinguish two classes of attributes that we call background and real-time attributes. *Background attributes* are the relatively static observations which practically don't change their values during message transmission (e.g., the job of the message sender). Conversely, *real-time attributes* (e.g., sender geographic localization) are dynamic observations which can take new values at run-time and thus might trigger changes in the routing policy behavior. Background and real-time attributes can be interpreted or not. The *context aware adaptive routing policy* constitutes the core of the proposed model. It clearly shows the dependency, as well as the sensitivity of the routing policy applications to context information stemming from source, target and message ecosystems.

IV. MODEL-DRIVEN ARCHITECTURE FOR CONTEXT-AWARE ROUTING POLICY MANAGEMENT

In this Section, we present our functional architecture (see Figure 2), which aims to be as generic as possible and to comply with different scenarios, with the objective to allow the steering of an intelligent, adaptable and flexible message routing policy. The architecture we propose is composed of five components we describe hereafter and it is mainly based on Service Oriented Architecture.

A. *Message reporter*

This module represents the source or the origin of the message. It is responsible for message reporting and publishing to the context-aware intelligent routing policy manager (RPM). It is usually a device (smartphone, PDA, PC, phone, etc.), controlled by a person or a software application.

B. *Ecosystem supervisor*

The main role of this module is to supervise and to listen continuously to the ecosystems. It compares real-time and background context attributes, detects changes and notifies the RPM. Notifying the RPM is triggered when observations differ from more than a given threshold from their previous values. It is like a contract carried out between the RPM and

 ISBN: 978-1-61208-405-3

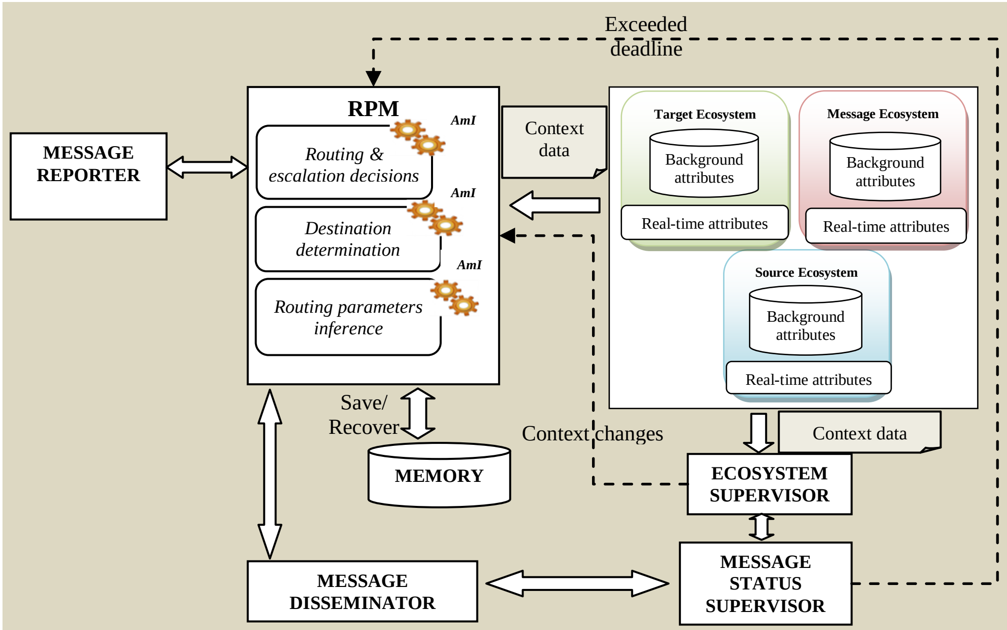

Figure 2. Functional architecture for context-aware routing policy management

the ecosystem supervisor to identify the situations under which the RPM requires behaviors adaptation.

C. *Message status supervisor*

This module supervises the status of the message. It shall detect deadlines exceeding of message reception, reading and reply, and notify the ecosystem supervisor and the RPM, which shall adapt its decision, if necessary. The message status supervisor activates its own timing system each time it receives from the message disseminator a notification indicating a message sending and the required delay for reading, reception and reply. To this aim, we can adopt a Timed Petri Net (TPN). Indeed, TPN is a convenient method to analyze and model the dynamics of self-adaptive systems [13]. Its main advantage comes from its expressiveness of timing constraints, such as durations of performances and deadlines. Several recent works [16] [17] highlighted some of the key features of TPN, such as safety, liveness and reachability. In our case, the message chain history can be designed as a TPN model where place consists of the message state (waiting for reception/ for read/ for reply) and token represents the message itself. The admissible duration of a message stay in a given place corresponds to required delay already determined by the routing policy manager (see Section E). We can take advantage from TPN at two levels: on one hand, it avoids message deadlocks, thus enabling the possibility of message redirection; on the other hand, TPN participates in the determination of the destination's trust level, which is considered among the most important criteria to take into account when identifying relevant destinations.

D. *Message disseminator*

It receives the message from the RPM and dispatches it to the destinations endpoint devices. It reports also the communication message status to the RPM and notifies the message status supervisor once the message is sent.

E. *Routing policy manager (RPM)*

The RPM constitutes the core of the architecture. It interacts with all the rest of the modules. The RPM is mainly based on ambient intelligence technology, which will allow to reason about the context in order to enhance the quality of message transmission. The module will proceed at operational level and include real-time decisional processes that will react to context, as well as to unpredictable events, such as destination unavailability, localization change, etc. The RPM has several roles where each one is played by a service component in the RPM structure. The basic responsibilities are as follows:

- Receive messages coming from the message reporter.
- Report the message status to the message reporter.
- Collect contextual information from the source, message and target ecosystems.
- Analyze data to make them acceptable and exploitable for interpretation.
- Infer routing parameters: For each scenario, the RPM determines the requirements and the appropriate parameters essential to route the message to the relevant destination.
- Assign the demand to the appropriate destination.
- Dispatch the message to the message disseminator.
- Cancel message routing by asking the message disseminator for stopping dissemination in order to redirect the message to another destination, if necessary.

 ISBN: 978-1-61208-405-3

- Save/recover contextual data in/from the cache memory.
- Delete messages from the memory in case of request from the message reporter or disseminator (e.g., because of obsolete or undeliverable messages).

The routing policy manager must face two major challenges: (1) to well understand the context information; (2) to dynamically adapt its behavior at run-time according to the context. In our design, the routing policy adaptation will operate at three levels, each of which constitutes an RPM service that we describe hereafter.

1) Routing and escalation decision: At run-time, an escalation (escalate, descalate) and/or a routing decision ((re)send, redirect, cancel) may be triggered/modified, depending on the ecosystem supervisor, message status and message disseminator results notifications.

For instance, the service can change/modify the message itinerary because of changes in the destination context (e.g., geographic localization) or for exceeding the required message reply delay detected by the message status supervisor, or because of connection problems identified by the message disseminator. In case of message redirection, the routing and escalation service calls the routing parameters inference service to determine new requirements, viz a new typical destination profile.

2) Routing parameters inference: The mission of this service is to reason about context information belonging to the message and source ecosystems in order to determine the message routing parameters, already defined in the previous Section. The routing parameters inference service determines, for each scenario, a destination profile type, as well as the required delays for message reception, reading and reply.

For that purpose, classical rule based techniques can be used to infer routing parameters in function of the context information. Such an approach however, although appropriate for static applications, is rather difficult to set up for dynamic applications. Also, building the rules set requires to predict all possible context configurations, which is not so evident.

For solving the routing parameters inference problem, we have two dimensions to take into account: *(1) Time* and *(2) uncertainty*.

a) Time: In some scenarios, at run-time, the message can follow multiple routing policies. The routing parameters and the observations at time t may depend on routing parameters and observations at previous time t-1.

b) Uncertainty: To reason about context information, the routing parameters inference service must be able to deal with uncertainty. Indeed, according to the sources they come from (e.g., noisy sensors), context information can be uncertain, incomplete or imprecise.

Hence, we have to find the appropriate tool that is able to model the routing policy dynamicity in function of time and uncertainty. The objective is to make the routing policy intelligently adaptive according to observations evolving over the time.

Dynamic Bayesian Networks (DBNs) [18] may be suitable for that. A key feature of a DBN is to unify the representation of temporal dimensions and of uncertainty. A DBN is a Bayesian Network which relates variables to each other over adjacent time steps called time slices. These temporal connections incorporate conditional probabilities between variables based on the Markovian condition that the state of the system at time t depends only on its immediate past, i.e., its state at time t-1. Based on the stochastic formalism, DBNs allow to infer the probability of unknown states, given some known observations and the initial probability distributions. Initial probabilities may be computed on the basis of experimental data with machine learning technics. Probabilistic inference is defined as the process of deriving logical conclusions from known, or assumed to be true, premises. The problem of inference in DBN consists in finding $P(X^{t-1}|Y^{t-1})$, where Y^{t-1} represents a set of t consecutive observations, and X^{t-1} is the set of the corresponding hidden variables. Forward-backward and junction tree algorithms are some examples of inference algorithms that may be used in DBNs.

In our case, we can imagine using a DBN to infer the probability of the message routing parameters as hidden variables, based on observations coming from the source and message ecosystems. For instance, a DBN might be used to estimate the probability level of the requested destination staff type (medical, rescue, assistance, etc.) and use it as a routing parameter, given the source ecosystem observations (message expeditor trust level high/medium/low), the message subject importance level (high/medium/low), and the message ecosystem observations (informative/alarm message type, emergency level high/medium/low, etc.).

3) Destination determination: This service addresses the following question: for a given scenario, to which relevant destination the message must be sent? Using a set of preconfigured recipients, the destination determination service shall search for the nearest destination that is closest to the typical destination profile which has already been determined by the routing parameters inference service. This process needs beforehand an in-depth analysis of the target ecosystem context. To select the relevant destination, several methods can be adopted such as multi-criteria utility function or K-Nearest Neighbor algorithm (K-NN) [19].

V. USE CASE: MESSAGE ROUTING ADAPTATION IN HEALTHCARE APPLICATION

In this Section, we illustrate the need for implementing dynamic message routing policy management applications with a scenario stemming from the healthcare domain. Obviously, healthcare applications are both mission-critical and real-time since they require in-time responses especially in emergency cases. The following scenario shows how the RPM will adapt its behaviors according to context changes belonging to the source ecosystem.

Scenario: Patient A has a history of cardiac disease. He visits a high mountains area for skiing, taking his intelligent cardiac device with him. While skiing, he felt a chest pain. His care device reports an alarm message to the RPM. In this case, the care device operates as a message reporter. The latter sends the message to the RPM which confirms the message reception. The RPM collects as a first step interpreted and/or non-interpreted, background and/or real-time contextual data from the source ecosystem (e.g., interpreted real-time attribute: environment type: hostile; interpreted background attribute: history cardiac problem: yes) and from the message ecosystem (e.g., interpreted real-time attribute: message importance level). The RPM unifies the collected observations and calculates the initial probability distribution necessary for the DBN. The latter infers a typical profile satisfying such criteria, e.g., staff type: medical, the required resources materials (viz an helicopter because of hostile patient A environment) and the delays for message reception, reading and reply. In a preconfigured destination list, the RPM searches a profile that is nearest to the typical profile and associates the message to patient A's admitting physician (physician B). The message disseminator sends the message to physician's B PDA. Physician B confirms message reception within the specified delay; however he exceeds the delay required for replying. In between, the patient' chest pain has become more acute. The ecosystem and message status supervisors notify the RPM of these changes. New observations are notified to the RMP which adapts its decisions by escalating the message priority and redirecting it to an emergency department as a new destination, which thus takes care of the patient. Let us note that a history of the exceeded deadlines and negative responses may decrease physician B's trust level which is considered an important criteria to take into account when choosing the relevant destination. Indeed, some specific context situations and reasons for which a physician may reject a healthcare request shall also be taken into account. For example, physician B can be unavailable because of commitment in another task, of vacations, etc.

VI. CONCLUSION

In this paper, we present a model and an architecture that emphasize the multidimensionality of the message routing problematic. Our objective is to ensure QoS of message transmission in terms of delivering the message to the right destination with respect to the required delays for message reception, reading and reply, and taking into account message, source and target context changes. Meanwhile, we believe that developing an intelligent adaptive routing policy in pervasive environments may save lives and money, especially in emergency scenarios. Within the proposed architecture, we also highlighted three adaptive services for which we propose appropriate methods to make them sensitive to context changes and to exceeded message deadlines. One of the major assets of our data and model driven architecture approach is to embed artificial intelligence methods at different levels: to infer routing parameters, to choose the right destination, and to select relevant context data thus avoiding the need for big data exchange.

ACKNOWLEDGMENT

This research work was supported by grants from AXON' SAS F-51210 MONTMIRAIL France and ANRT.

REFERENCES

[1] A. Saeed and T. Waheed, "An extensive survey of context-aware middleware architectures," IEEE International Conference on Electro/Information Technology (EIT 2010), IEEE Press, May 2010, pp. 1–6, doi: 10.1109/EIT.2010.5612118.

[2] M. Salehie and L. Tahvildari, "Self-adaptive Software: Landscape and Research Challenges," ACM Transactions on Autonomous and Adaptive Systems, vol. 4, May 2009, pp. 14:1–14:42, doi: 10.1145/1516533.1516538.

[3] Y.-F. R. Chen and C. Petrie, "Guest editor's introduction - Ubiquitous mobile computing," IEEE Internet Computing, vol. 7, Mar. 2003, pp. 16–17, doi: 10.1109/MIC.2003.1189184.

[4] C. Taconet, Z. Kazi-Aoul, M. Zaier, and D. Conan, "CA3M: A Runtime Model and a Middleware for Dynamic Context Management," in On the Move to Meaningful Internet Systems, vol. 5870, R. Meersman, T. Dillon, P. Herrero, Eds. Heidelberg, 2009, pp. 513–530.

[5] C. Ramos, J. C. Augusto, and D. Shapiro, "Ambient Intelligence-the Next Step for Artificial Intelligence," IEEE Intelligent Systems, vol. 23, Mar. 2008, pp. 15–18.

[6] Y. Sun, W. Tan, L. Li, G. Lu, and A. Tang, "SLA detective control model for workflow composition of cloud services," IEEE 17th International Conference on Computer Supported Cooperative Work in Design (CSCWD 2013), IEEE Press, Jun. 2013, pp. 165–171, doi: 10.1109/CSCWD.2013.6580957.

[7] R. Ding and G.-M. Muntean, "A context-aware cross-layer energy-efficient adaptive routing algorithm for WLAN communications," IEEE 37th Conference on Local Computer Networks (LCN 2012), IEEE Press, Oct. 2012, pp. 176–179, doi:10.1109/LCN.2012.6423600.

[8] S. Peng, S. X. Yang, S. Gregori, and F. Tian, "An adaptive QoS and energy-aware routing algorithm for wireless sensor networks," International Conference on Information and Automation (ICIA 2008), IEEE Press, Jun. 2008, pp. 578–583, doi: 10.1109/ICINFA.2008.4608066.

[9] F.-S. Hsieh, "Context-aware Workflow Driven Resource Allocation for e-Healthcare," 9th International Conference on e-Health Networking, Application and Services (HealthCom 2007), IEEE Press, Jun. 2007, pp. 34–39, doi: 10.1109/HEALTH.2007.381599.

[10] N. B. Behlouli, C. Taconet, and G. Bernard, "An architecture for supporting Development and Execution of Context-Aware Component applications," ACS/IEEE International Conference on Pervasive Services (ICPS 2006), IEEE Press, Jun. 2006, pp. 57–66, doi: 10.1109/PERSER.2006.1652207.

[11] Y. Oh, J. Han, and W. Woo, "A context management architecture for large-scale smart environments," IEEE Communications Magazine, IEEE Press, vol. 48, Mar. 2010, pp. 118–126, doi: 10.1109/MCOM.2010.5434383.

[12] M. Miraoui, C. Tadj, J. Fattahi, and C. Ben Amar, "Dynamic Context-Aware and Limited Resources-Aware Service Adaptation for Pervasive Computing," Advances in Software Engineering, vol. 2011, Feb. 2012, pp. 1-11.

[13] N. Cardozo, S. Gonzalez, K. Mens, R. Van Der Straeten, and T. DHondt, "Modeling and Analyzing Self-Adaptive Systems with Context Petri Nets," International Symp. on Theoretical Aspects of Software Engineering (TASE 2013), IEEE Press, Jul. 2013, pp. 191–198, doi: 10.1109/TASE.2013.33.

[14] F. Fleurey, V. Dehlen, N. Bencomo, B. Morin, and J.-M. Jézéquel, "Modeling and Validating Dynamic Adaptation," in Models in Software Engineering, vol. 5421, M. R. V. Chaudron, Eds. Heidelberg, 2009, pp. 97– 108.

[15] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a Better Understanding of Context and Context-Awareness," Proc.1999 Handheld and

Ubiquitous Computing Symp. (HUC 1999), Springer-Verlag, 1999, pp. 304–307.

[16] Z. Hu, T. Lu, and Z. Zhao, “Context-aware service system modeling using timed CPN,” 10th International Conference on Service Systems and Service Management (ICSSSM 2013), IEEE Press, Jul. 2013, pp. 164–169, doi: 10.1109/ICSSSM.2013.6602579

[17] R. B. Dilmaghani and R. R. Rao, “Supervisory decision making in emergency response application,” IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops 2013), IEEE Press, Mar. 2013, pp. 553–558, 10.1109/PerComW.2013.6529557.

[18] Z. Ghahramani, “Learning dynamic Bayesian networks,” in Adaptive Processing of Sequences and Data Structures, vol. 1387, C. Lee Giles, M. Gori, Eds. Heidelberg, 1998, pp. 168–197.

[19] G. Gutin, A. Yeo, and A. Zverovich, “Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP,” Discrete Applied Mathematics, vol. 117, Mar. 2002, pp. 81–86.