Performance Evaluation of Distributed M3 Applications via ABSOLUT

Subayal Khan, Jukka Saastamoinen, Jussi Kiljander, Jyrki Huusko, Juha Korpi
VTT Technical research Center of Finland,
FI-90570, Oulu, Finland
email:{subayal.khan, jukka.saastamoinen, jussi.kiljander, jyrki.huusko, juha.korpi}@vtt.fi

Jari Nurmi
Tampere University of Technology,
Department of Computer Systems
P.O.Box 553 (Korkeakoulunkatu 1),
FIN-33101 Tampere, FINLAND
jari.nurmi@tut.fi

***Abstract*—For future smart applications running on nomadic embedded devices operating in smart environments, the information sharing without human involvement, availability of the services and the quest for better alternative services will go hand in hand to enable the best-possible experience for the end-user. These tasks are challenging since the devices operating in a smart environment are usually heterogeneous and targeted for different uses. M3 is a layered architecture which aims at providing interoperability between heterogeneous devices in a Smart Space. M3 divides the interoperability challenge into three layers, i.e., information, service and communication level. The information level ensures that the information is understood in the same way by all devices. The service level guarantees the seamless access to services and discovery of new services while the communication level provides means for data transmission among devices. The design and development of new M3 applications is challenging not only due to various application design alternatives but also due to many solutions for achieving interoperability at different M3 levels. Therefore, a brisk performance evaluation phase is required for evaluating the feasibility of new M3 applications. In other words, the methodology should not only provide the feasibility of information, service and device level solutions employed by the instantiation of M3 but also the feasibility of the M3 applications on various platforms. This article describes an approach for the design and system level performance evaluation of M3 systems via UML2.0 MARTE profile and ABSOLUT and is experimented via a case study.**

Keywords- M3; SystemC; Performance Simulation

I. INTRODUCTION

Distributed embedded systems involve collaboration among connected embedded devices for example mobile handheld devices, embedded PC's, wireless sensors, robotics systems and microcontrollers. By accessing and making use of information from other sensors and embedded devices on the same network, rich scenarios can be created in different application domains of distributed systems. The overall value of the embedded sector worldwide is about 1600 billion Euros (€) per year. The three main market segments for embedded systems are telecommunications, automotive, and aerospace. The combined value of these segments is 1,240 billion € per year and are growing at a rate of over 10 per-cent [1].

In order to realize the full potential of this market segment, the efforts of industry and research communities are focused at the development of new pervasive standards and protocols for communication and information sharing among networked devices. These technologies will enable the services and software entities hosted on the devices in a pervasive environment to collaborate efficiently for information sharing to enable a richer end-user experience.

The networks of such devices are generally called smart spaces. The smart spaces are based on the concepts which are strongly correlated with those employed in the area of pervasive computing or ambient intelligence [2][3].

The devices from different domains which are a part of smart space can share information if they can interoperate. Currently several domain and vendor specific solutions of interoperability exist such as UPnP in the entertainment domain and Apple ecosystem controlled by a single vendor. These domain and vendor specific solutions, generally, do not allow the devices from one vendor and domain to interoperate with devices from other vendors or domains. The only way a device can overcome this barrier currently is to implement several different standards, which will enable it to participate in the use inter-domain and inter-vendor use-cases. Also, the existing standards often target specific use cases rather than attempt to specify a general framework for interoperability [4].

M3 (Multi-vendor, multi-device, multi-domain) is a generic interoperability framework for smart environments. It is semantic information sharing solution for smart environments which instantiates information level on top of NoTA (Network on Terminal Architecture) or any other service level interoperability solution [4]. This layered architecture allows for the separation of concerns at each layer. The concerns at each layer can vary between M3 applications. For example, an M3 application can demand both service access (over various transport technologies) and common service discovery mechanisms. In order to achieve these two objectives, an M3 service level IOP (interoperability) solution (such as NoTA) can be used to enable seamless service access while an M3 information level IOP solution, such as RIBS (RDF (Resource Description Framework) Information Base Solution), can be used for service discovery [5].

NoTA implements the multi-transport mechanism in the form of a DIP (Device Interconnect Protocol) that abstracts away the complexity and algorithmic details involved in service access over multiple transports and provides a simple modified Socket API (application interface) to application programmer collectively called NoTA BSD-SOCKET API functions. In this article, we utilize the M3 information level for enhancing the service discovery in NoTA based M3 applications (employing NoTA as service level IOP). This is achieved by presenting semantic descriptions of the functionalities provided by NoTA services. This allows the clients to find the suitable services more efficiently.

Deployment of new M3 applications is challenging not only due to the heterogeneous parallelism in the modern mobile platforms, but also due to performance and energy constraints. For efficient development and deployment of

M3 applications, it is of pivotal importance that the application design phase acts as a blue print for the SLPE (system-level performance evaluation) approach. Another major requirement is that the SLPE approach must be able to identify the potential bottlenecks at all the M3 levels in the M3 instantiation employed. In other words, before the deployment of an M3 application, the performance of IOP solutions operating at each level of M3 must be evaluated. An instantiation of M3 means the deployment of a specific interoperability solution at each M3 layer.

For SLPE of M3 applications, Abstract workload based performance simulation (ABSOLUT) has been used. ABSOLUT is a Y-Chart [6] based system level performance simulation approach consisting of application workload and platform model [7]. ABSOLUT employs model-based approach for applications, platform services, middleware technologies and platform [7].

So far, ABSOLUT has not been used for the SLPE of distributed M3 systems, i.e., the systems which comprise of multiple devices operating in a smart environment. Performance evaluation of distributed M3 applications via ABSOLUT demands the design, implementation and integration of information, services and communication level M3 models into the ABSOLUT methodology.

The first novel contribution of this article is to elaborate the design of M3 applications at service and information level via UML 2.0 MARTE profile [32]. The NoTA application (operating at the M3 service level) was previously presented in the case study described in [8]. In this research article, an additional application view was provided to represent the M3 information level in the NoTA application model. Also SSAPI (Simple Sockets Application Interface) was used for sharing information among information level M3 entities like KPs (Knowledge Processor) and RIBS. In other words, modelling of M3 applications employing NoTA is achieved by defining the information level in a separate application view within the overall NoTA application model. The rest of the application views represent the service level components of the software system.

The second contribution is the design and integration of information level protocol models to the ABSOLUT framework. The extended ABSOLUT framework is applied for the SLPE of the distributed M3 application presented in the case study. The framework can be used to evaluate the performance of IOP solutions operating at each of the three levels of M3.

The rest of the paper is organized as follows: Section II first gives a brief outline of landmark performance simulation and application modelling tools and techniques. Afterwards, it provides an overview of NoTA and M3 technologies. Section III describes the way M3 based smart space applications can be modelled. The section describes different application views and the way non-functional properties are carried through application modelling phase. Section 4 describes the modelling of M3 applications via a case study. Section 5 describes the performance modelling approach. Section 6 describes the performance modelling of the application described in Section 4. Section 7 first elaborates the system level performance evaluation results of the application, i.e., the way non-functional properties of the application are validated. This section also shows the performance of components in the platforms on which the application components were mapped. Afterwards, it describes the performance of IOP solutions operating at each of the three levels of M3. Conclusions and future work are outlined in Section 8 followed by acknowledgements and list of references.

II. Related Work

A detailed survey of the salient system level performance evaluation methodologies is provided in. Therefore, in the current section, we first provide a brief overview of these methodologies/tools. This is followed by a description of the M3 Framework and NoTA SOA (service-oriented architecture).

A. Existing Application modelling Tools/Languages

Object Management Group (OMG) defines Model Driven Application Architecture (MDA) relying on efficient use of system models to facilitate transformations between different model types. Various Architectural Description Languages (ADLs) have been proposed. MBASE provides integrated models for capturing the product success, process and properties [10]. ACME relies on a core ontology comprising of seven elements representing architectural elements [11]. MAE [12] triggers the modelling, analysis, and management of different versions of architectural artefacts supporting domain-specific extensions to capture other system properties.

B. Existing System Level Performance Evaluation Techniques

Performance modelling has been approached in different ways. SPADE [13] treats applications and architectures separately via a trace-driven simulation approach. Artemis [14] extends SPADE by involving virtual processors and bounded buffers. The TAPES [15] abstracts functionalities by processing latencies covering the interaction of associated sub-functions on the architecture without actually running application code. ABSOLUT [7] is system level performance evaluation for embedded systems which employs model based approach for both application and platform.

C. M3 framework

The aim of M3 is to provide multi-device, multi-domain and multi-vendor interoperability by combining Semantic Web technologies with publish/subscribe-based interaction. The interoperability challenge in M3 is divided into three levels: communication, service and information. The basic principle of M3 is that the information level interoperability is achieved by agreeing on common ontology models. On the communication and service levels, M3 relies on existing solutions. In this work, we utilize the NoTA technology to provide interoperability in the lower levels.

In M3, the W3C's (World Wide Web Consortium) Semantic Web specifications, such as RDF [33], RDFS (RDF Schema) [34], and OWL (Web Ontology Language) [35], provide the key technologies for the ontology-based interoperability. The RDF is a W3C standard designed to represent Web resources in a structured manner using subject, predicate and object triples. RDFS and OWL in turn provide vocabularies on top of RDF to describe any information as machine-interpretable ontologies. In M3, these technologies are exploited for representing information about the real world in order to create location-aware services to physical places.

The M3 functional architecture defines two types of entities: KP and SIB (Semantic Information Broker). KPs are software agents that provide the end-user with services by interoperating with each other. SIB is a shared blackboard providing methods for KPs to share machine-interpretable data in the smart space. The publish/subscribe based SSAP (Smart Space Access Protocol) defines the rules for KP-SIB communication. The SSAP provides following operations: *join()*, *leave()*, *insert()*, *remove()*, *update()*, *query()*, *subscribe()*, and *unsubscribe()*. The Fig.1 illustrates how the SIB and KPs form M3-based smart spaces.

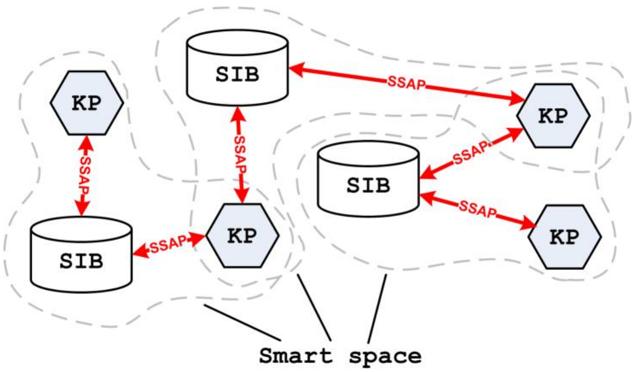

Figure 1. Composition of M3-based smart spaces.

Two versions of the M3 concept have been published: Smart-M3 and RDF Information Base Solutions (RIBS) [5]. Smart-M3 is the first official implementation of the M3 concept. It is a Linux based implementation that utilizes XML (Extensible Markup Language) serialized format of the SSAP. Smart-M3 provides both NoTA and plain TCP/IP based communication technologies and it has been also implemented as a service in OSGi (Open Services Gateway initiative) framework [16]. The Smart-M3 supports two types of query formats: simple template queries and WQL (Wilbur Query language) [17]. In template query the query string consists of separate RDF triples, which are matched against the RDF database of the SIB. The WQL query in turn consists of start node and a path to be traversed from the start node.

The RIBS is an ANSI-C implementation of M3 concept designed for portability, security, and performance. Similarly to Smart-M3, it supports both plain TCP (Transmission Control Protocol) and NoTA based transports. In contrast to Smart-M3 the RIBS uses WAX (Word Aligned XML) serialized SSAP format. The WAX serialized SSAP messages are more compact and faster to parse and therefore provide better performance and portability to low capacity devices. The query languages supported by the RIBS are also bit different from the Smart-M3.RIBS does not support WQL but provides limited support for SPARQL [5]. TLS (Transport Layer Security) and RDF-triple level access control mechanisms are used to provide security and privacy in smart spaces.

M3 concept is based in the voluntary sharing of information by objects in physical space. It is solely up to information owner to decide what and how information is published. M3 ensures the availability of information from physical world to devices and novel applications in smart environments. In this way the applications can enhance end-user experience by taking advantage of the available information in the smart space and by creating create new cross-domain use cases. In this article, we use M3 to enhance the service discovery of NoTA services available in the smart environment.

D. *Network on Terminal Architecture (NoTA)*

NoTA is a novel SOA which consists of three types of logical elements: SNs (Service Node), ANs (Application Node) and DIP (Device Interconnect Protocol). Service nodes are services that can be used by ANs and other SNs. Application nodes are the application functionalities composed of service calls and other logic. Communication between the Application and Service Nodes takes place always over the DIP.

The DIP defines both types of socket based communication, i.e., it supports both message and streaming type of data flows. NoTA DIP is divided into two main functional blocks. The first one is called H_IN, which manages service registration, discovery, access and security. The second is called L_IN, which is responsible for connecting the subsystems together.

From a software architect's perspective, the applications supported by NoTA systems are modelled as NoTA SOA [9]. In other words, a NoTA application consists of a set of Application Nodes (ANs) and Service Nodes (SNs) which collaborate via NoTA Device interconnect protocol to satisfy a use-case. For modelling a novel SOA for embedded nomadic devices (in this case NOTA SOAD), UML 2.0 MARTE profile comes as a natural choice [9].

III. MODELLING NOTA BASED M3 APPLICATIONS

In this article, we focus on the design of M3 applications which employ RIBS and NoTA at information and service level respectively.

NoTA based M3 applications are those M3 applications which use NoTA as a service level IOP. From this point onwards, we use the term "M3 applications" for these applications. It should be noted that the same modelling languages and techniques can be employed for M3 applications based on other serviced level IOP, such as ADIOS (Adaptive Input/Output System) and OSGi. In each case, the SOA concepts are used with additional view(s) representing the information level.

It should be noted that, since the M3 information level is instantiated on top of service level, therefore, each information level software entity is also represented as an entity at the service level. In case of Smart M3, the information level entities are KPs and SIB/RIBS while the service level (generally speaking) entities consist of servers and clients (called SNs and ANs in NoTA SOA).

Therefore, in case of NoTA based M3 systems, each KP is an AN at the service level and each SIB/RIBS is a SN at the service level. After M3 application design, the software components representing service and information level application components (KPs, RIBS, ANs, and SNs) and technologies are mapped to platforms to constitute the complete M3 system. In the next subsections, we elaborate the modelling of a complete M3 application (employing NoTA as service level IOP).

a. *M3 application views*

The M3 application modelling process starts by describing a set of views that are sufficient for the modelling objective. These views are instantiated by using UML2.0 MARTE profile and are illustrated in conjunction with the RM (Restaurant Multimedia) Application case study. The use case view describes the functionality of a system at a higher abstraction level by means of use cases. The structural view defines the interface between an application and the sub-systems of the execution platform.

 ISBN: 978-1-61208-246-2

The interfaces are implemented by the ANs and SNs. The syntactical view describes the syntax of the messages passed between ANs and SNs. The behavioural view reflects the behavioural aspects of an application and its encompassing services. All the views described so far describe the service level of the modelled application. The information level is described in a separate view, called the semantic view, which describes the semantic description of the information used by the information level entities called KPs and RIBS/SIB. The KPs are the ANs, which use the information contained in RIBS/SIBS which in turn are SNs at service level.

b. Non-Functional Properties

In case of M3 applications, the end user experience can be affected by the IOP solutions operating at device, service and information level. Therefore, the performance of IOP solution at each layer of M3, as well as the application, must be analysed to identify the potential bottlenecks.

It means that in case of NoTA based M3 systems, apart from the end-end delays of messages exchanged among devices, the processing times of NOTA DSD (Data Structure Diagram) API functions and SSAP API functions are also important. Hence we will employ ABSOLUT for analysing the following non-functional properties.

- The processing times of the targeted application functions and external libraries.
- At the M3 device level, we will analyse throughput at MAC-Level, throughput at Transport-Level, Average Frame Delays, Average Transport Delays, Frame loss rate and Packet Loss rate.
- Processing times of the NoTA API functions.
- Processing times of SSAP functions which gives a good insight into the performance of RIBS (Information level IOP).

The non-functional properties of an M3 application (from the end-user perspective) are identified and elaborated in the syntactical view. Firstly they are shown in the extended behavioural view and later on validated by the performance simulation. We focus in the sequel on one non-functional property, FrameRate, showing the way it is carried through the design process for the design of a certain NoTA SN in the distributed M3 application at service level. This is outlined in Figure 2.

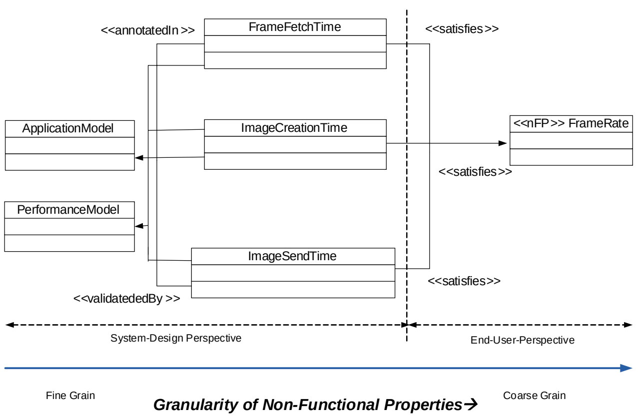

Figure 2. Carrying FrameRate through the application modelling and performance simulation process

IV. CASE STUDY: MODELLING RESTAURANT MULTIMEDIA M3 APPLICATION

We now describe the modelling of a RM M3 Application. This application allows a customer to request his preferred multimedia service and is hosted on his mobile device. At the service level, the application is like a control, which can request any of the three Application Nodes (ANs) for a specific functionality, i.e., for viewing a News Channel, a music video or a movie. Each of these ANs then requests its corresponding Service Node (SN) to access the streaming multimedia content. The nodes, and their required and provided interfaces, are elaborated in the application model.

The information level is expressed in the Semantic view which illustrates the information regarding the available services in the smart space. This information is used by the mobile devices of customers entering the restaurant to avail the desired multimedia services.

A. Application use-case view

The use case view shows a system level capability i.e., selection of Multimedia Service, as shown in Figure 3.

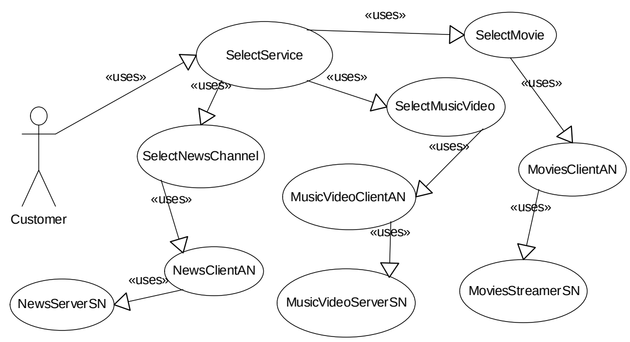

Figure 3. Restaurant Multimedia application use-case view

***a.** Application syntactical view*

The syntactical view describes the syntax of messages passed between the ANs and application and also between ANs and SNs. This is shown in Figure 4.

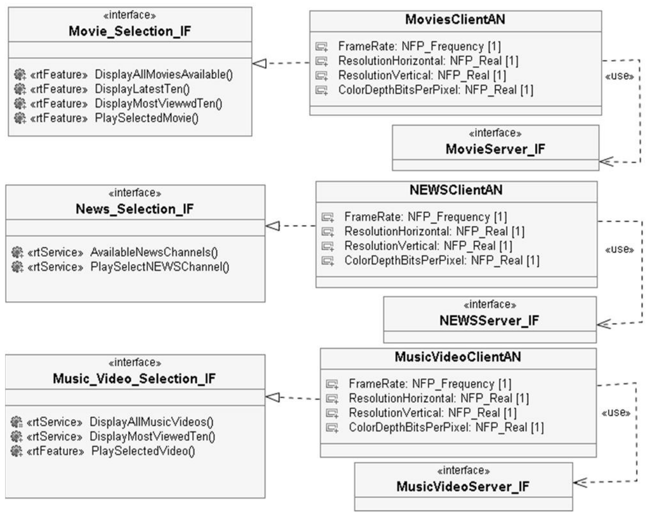

Figure 4. Diagram showing Interfaces realized and required by ANs.

The interfaces needed by ANs are provided by SNs and are shown similarly. The NFPs (non-functional properties) [18] are assigned values in respective slots of their instances and are shown in Figure 5.

Figure 5. The non-functional properties represented as slot values

b. Application Semantic View

The semantic view illustrates the semantic descriptions of SNs used in the discovery process. The ontology suite designed for the discovery consists of two ontologies: NSO (NoTA Service Ontology) and VSSO (Video Stream Service Ontology). Figure 6 illustrates these ontologies as an RDF graph.

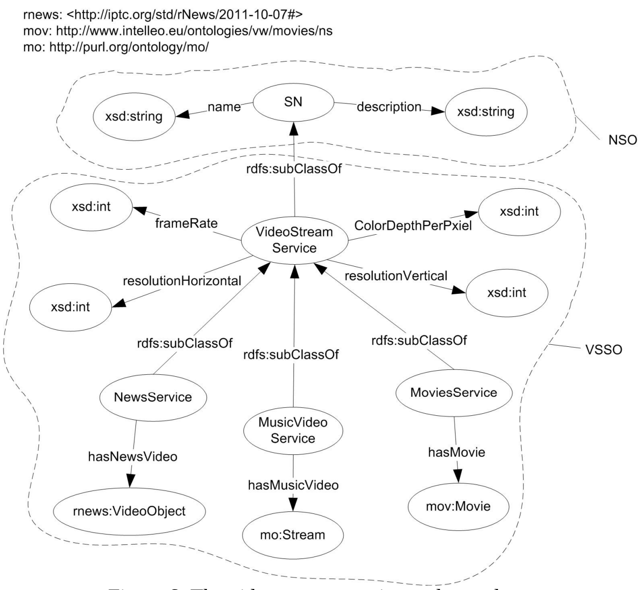

Figure 6. The video stream service node ontology

NSO is general purpose ontology for describing NoTA services. It contains just a one class (ServiceNode), which presents common information such as the name and the human readable description about the NoTA service. The idea is that when new NoTA services are designed for specific application domains, the domain specific ontologies import the NSO ontology and introduce new subclasses for the common ServiceNode class.

The purpose of the VSSO is to describe the capabilities of video stream services in a machine-interpretable format. The VideoStreamService class is the main class of the ontology. It is defined as a subclass of the ServiceNode class of the NSO ontology, but it is also possible to utilize it with different SOA technologies. The VideoStreamService class contains properties such as frameRate, resolutionHorizontal, resolutionVertical, and ColorDepthPerPixel for describing the non-functional properties of the video stream service. By querying the values for these properties form the SIB, the ANs are able to select the service that best meets their requirements. The VSSO contains also three subclasses for the VideoStreamService class: NewsService, MusicVideoService, and MovieService. These classes represent services that provide specific types of video streams and contain properties such as hasNewsVideo, hasMusicVideo, and hasMovie for representing the actual video streams provided by the service. To provide suitable ranges for these properties we have imported following ontologies: rNews [19], Music Ontology [20] and IntelLEO [21] Movies Ontology.

c. Application Behavioral View

The behavioral view shows the behavior of the application as shown in Figure 7. First of all, the end-user application becomes aware of the services available in the smart environment by communicating via RIBS using SSAP API functions. This is called the service discovery phase.

Once the end-user application knows about the available multimedia services, it allows the user to avail the desired services. The functionalities of the multimedia services are implemented by (allocated to) NoTA ANs and SNs. These ANs and SNs must satisfy the non-functional properties annotated in the syntactical view for a better end-user experience. These non-functional properties are refined to a set of non-functional properties from the implementation perspective as shown in Section 5 and are validated by the performance simulation phase as shown in Section 6.

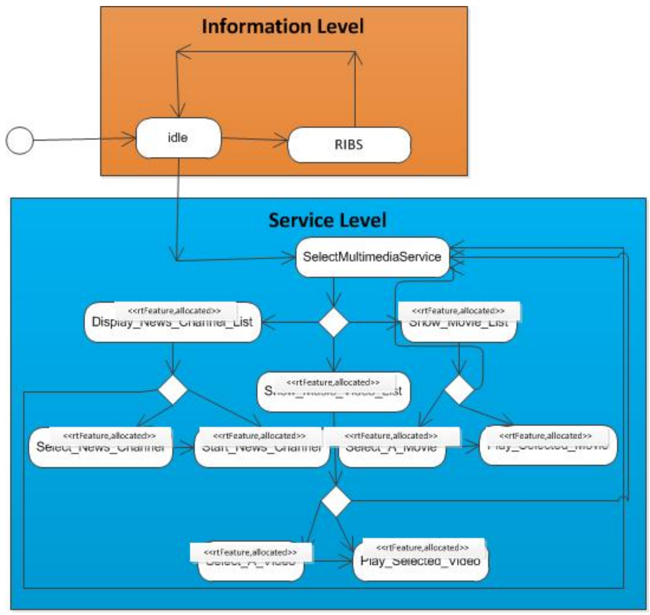

Figure 7. Behavioural view of Application.

V. PERFORMANCE MODELLING APPROACH

The performance modelling of NoTA based M3 systems requires the modelling of NoTA SOA workload models (workload models of ANs and SNs), modelling of SSAP API, KPs and RIBS at the information level, modelling of NoTA DIP operating in different modes and modelling of device level protocols/services, for example transport protocols, such as TCP/IP and UDP. We now describe the performance modelling of the aforementioned NoTA based M3 system in ABSOLUT.

A. Modelling NoTA SOA workload models

In order to integrate the NoTA SOA to ABSOLUT [22] seamlessly, the behavioural view of NoTA application model is extended to form a layered hierarchical structure of applications as described in [8]. The corresponding layers in the application workload models are identified. In this way, the application model acts as a blue print for the application workload models, reducing the time and effort in the performance evaluation phase [8].

 ISBN: 978-1-61208-246-2

In case of NoTA application model, the behavioural view represents a use_case as a controlled collaboration of ANs or SNs. Therefore, $USE\_CASE = \{C, N_1, N_2, N_3,\ldots, N_N\}$, where N_i is an *AN* or *SN* and *C* represents the control of the application. Corresponding workload model layer is $USE\_CASE\_MODEL= \{C_{WLD}, NM_1,\ldots,NM_N\}$, where NM_i is the workload model of an AN or SN and C_{WLD} is the control. Each of these workload models is an Application-Level ABSOLUT workload model.

Each AN or SN contains a set of processes and control, i.e., $N_i =\{C_P,P_1,P_2,\ldots,P_N\}$, where P_i is the model of a single process in an *AN* or *SN*. The corresponding workload model layer is $NMi=\{C_{PM},PM_1,PM_2,\ldots,PM_N\}$, where PM_i is the model of the *ith* process model. The processes models of a single *AN or SN* communicate via ABSOLUT IPC models as elaborated in [23] and are scheduled by the ABSOLUT operating system model [7].

The processes of an *AN* or *SN* can call library functions, system calls and functions of user-space code. For communication with other processes, they can call BSD_API functions or make use of IPC. The corresponding Process workload models call Function workload models and workload models for external library functions obtained by ABSINTH-2 [24]. The BSD API functions are modelled as Transport Services registered to the OS models [25].

The control and the functionalities of the MusicVideoServerSN (which consists of a single process) are shown in Figure 8. The non-functional property i.e. FrameRate is assigned the required value (40 Frames/sec) in the model element representing MusicVideoServerSN in Figure 5. This non-functional property is further refined to three non-functional properties from the design perspective, i.e., FrameRetrievalTimeMax, ImageCreationTimeMax and ImageSendingTimeMax. These refined non-functional properties are annotated in the behavioural view to their corresponding functionalities, i.e., Get a Frame, Create Image and Send the Image. The OPENcv [26] library functions, i.e., cvQuerryFrame and cvCreateImage and user-space function SendImage, providing these functionalities are mentioned below the name of these functionalities. Each of these non-functional requirements are analysed in the performance simulation phase to check whether the required FrameRate has been achieved. Due to the pipelined nature of the functionalities, each of them has to be performed within 1/40 seconds (to fulfil the required frame rate). The function SendImage is a wrapper around the NoTA BSD API Hsend() function [27].

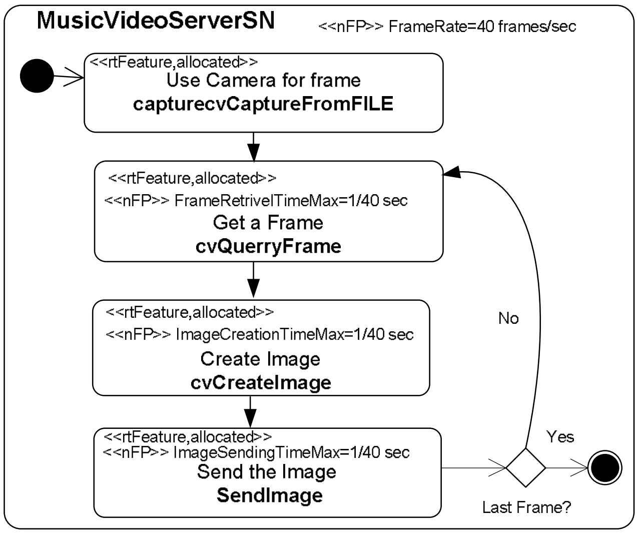

Figure 8. MusicVideoServerSN control with functionalities mentioning refined non-functional properties

Hence a single process of an AN or SN, "P_i", can be represented as $P_i=\{C_F,F_1,F_2,\ldots,F_N, S_1, S_2,\ldots S_K\}$, where F_i is a function and S_i is a service requested from platform. The corresponding workload model layer is $PM_i=\{C_{FM},FM_1,FM_2,\ldots,FM_N, SM_1, SM_2,\ldots SM_K\}$, where FMi is a function workload model and SMi is a platform service workload model. The mapping between the NoTA Application model layers and the corresponding Workload model layers are shown in Table I.

TABLE I. COMPARING NOTA APPLICATION MODEL LAYERS AND ABSOLUT WORKLOAD MODEL LAYERS

Application Layers	Workload Model Layers
use_case= $\{C,N_1,N_2,\ldots,N_N\}$	use_case_model=$\{C_{WLD},NM_1,.,NM_N\}$
$N_i = \{C_P,P_1,P_2,\ldots,P_N\}$	NMi=$\{C_{PM},PM_1,\ldots,PM_N\}$
$P_i=\{C_F,F_1,F_2,\ldots,F_N,S_1,\ldots S_K\}$	PMi=$\{C_{FM},FM_1,,FM_N,SM_1,\ldots SM_K\}$

a. Modelling SSAP API, RIBS and KPs

At the information level, the information repository and users, i.e., RIBs and KPs communicate via SSAP API. The workload models of RIBs and KPs are easily extracted via ABSINTH-2 [24] in exactly the same way as the ANs and SNs of NoTA SOA. The reason is that RIBS and KPs are nothing but SNs and ANs in terms of SOA (at M3 service level), which implement a specific functionalities, i.e., storing and using information using a protocol called SSAP API. SSAP API is in turn a set of wrapper functions over NoTA BSD API functions in case of NoTA based M3 systems. The workload models of SSAPI can be obtained via ABSINTH-2 [24].

b. Modeling NoTA DIP workload models

NoTA DIP is available as an external library and has also been implemented as platform service implemented in LINUX Kernel [8]. When used as an external library, NoTA DIP operates in two modes i.e., Single Process (SP) mode or Daemon mode [8]. In both cases, NoTA DIP services be requested by applications as modified NoTA BSD API functions. Linking NoTA Application architecture design to ABSOLUT demands the modelling of both NoTA implementations.

The design and integration of ABSOLUT workload models corresponding to different NoTA implementations (as services or external libraries) and operating modes (SP and daemon mode) are described in detail in [8].

 ISBN: 978-1-61208-246-2

c. Modelling Device Level Services

Modelling of device level services, for example transport protocols, such as TCP and UDP, is described in [25]. The MAC and transport layer models were compared to the corresponding models of widely used network simulators, i.e., ns-2 and OMNeT++ [29][30]. The results were 75-85% accurate as compared to these benchmarks and were always pessimistic. In other words, if the use-case requirements are validated by the ABSOLUT MAC and transport models, the results are surely validated by ns-2 and OMNeT++ simulators. The reason is that ABSOLUT models always give higher values of MAC and transport level delays and throughput under the same network conditions for example number of nodes and channel bit rate [25].

d. Overall M3 systems performance model

Therefore, the overall ABSOLUT performance model of an M3 system contains the hardware services, software services, platform components, the models of device level services**,** the ABSOLUT models of different NoTA DIP implementations and modes, the workload models of ANs and SNs extracted by extended application models and the workload models of SSAP API functions. Figure 9 shows the possible components of an overall ABSOLUT performance model of a M3 system. The ABSOLUT models corresponding to different M3 levels are shown in different colours in Figure 9.

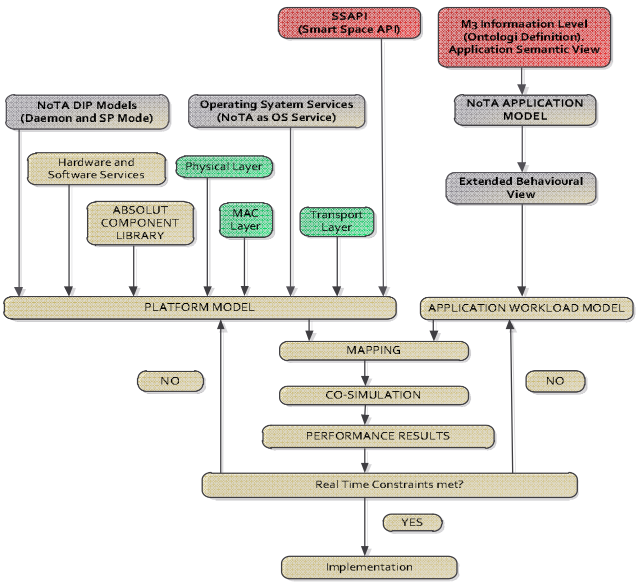

Figure 9. The modelled protocols/components of M3 in the corresponding ABSOLUT performance model

VI. PERFORMANCE MODELLING OF CASE STUDY

Each server and client (called SNs and ANs in NoTA) in real case study presented in Section 4 is modelled as a separate application-level workload model. Each Application-Level workload model of a NoTA AN or SN instantiates the process workload model mimicking its' execution in the real use-case. KPs and SIBs are also ANs and SNs, which store and share/use information about available services or contained in different devices. Hence the ABSOLUT workload models of SIBs and KPs are generated in the same way as other ANs and SNs [8]. Therefore, from this point onwards, we do not use the terms KP or SIB explicitly.

B. Overall ABSOLUT Performance Model

Each AN and SN presented in the application model is mapped to a separate ABSOLUT platform model to analyse the performance results and identify the potential bottlenecks at the software and hardware side. The overall performance simulation model is shown in Figure 10.

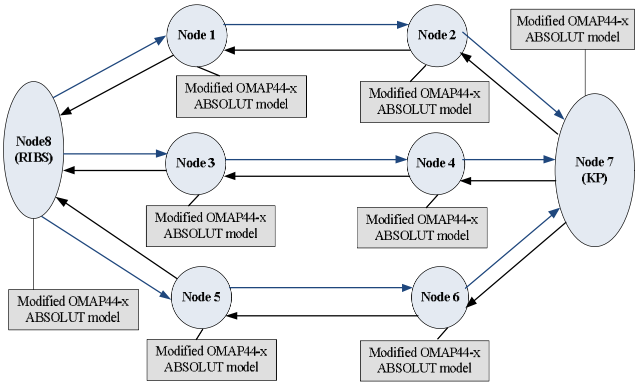

Figure 10. Performance model of the Restaurant Multimedia application

Node 1 and Node 2 represent the NewsServerSN and NewsClientAN. Node 3 and Node 4 represents the MovieStreamerSN and MovieClientAN, whereas Node 5 and Node 6 represent the MusicVideoServerSN and MusicVideoClientAN. Node 7 represents the application hosted on the mobile device of a customer entering the restaurant. The application is in the form of a control [22] and the user decides which services to use. Node 7 (the end-user application) first retrieves the information related to the available services (three multimedia services implemented by other nodes) while Node 8 represents SN implementing RIBS. Therefore, Node 7 acts as a KP. After knowing about the available services, the Node 7 contacts the related service on end-user's direction and the desired multimedia content is streamed to the customer's device.

It should be noted that the service nodes also communicate with the RIBS via SSAP to inform it about their presence. In this way, the information related to their capabilities and presence is made available to devices in the smart environment. The applications hosted by these devices can then use these services when desired by the end-user.

a. ABSOLUT Device Platform Models

Each ABSOLUT platform model used in the case study is a modified OMAP-44x platform model. It consists of two ARM Cortex-A9 processors consisting of four cores respectively instead of two (as in case of original TI OMAP44-x platform) along with SDRAM, a POWERVR SGX40 graphics accelerator and an Image signal processor. This is shown in Figure 11. The NoC infrastructure was abstracted out and replaced with on-chip bus as shown in Figure 11 . Each processor core (Cortex-A9 CPU model) has an L1 and L2 cache and can possibly share an L3 cache with one or more cores in the Multi-Core Processor model as described in [8].

 ISBN: 978-1-61208-246-2

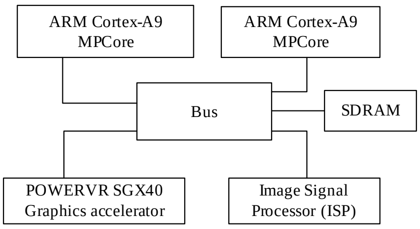

Figure 11. OMAP 44x Platform ABSOLUT model.

b. Application and M3 IOP workload Models

All the SNs (except RIBS) were programmed using OpenCV library [26]. The workloads of all ANs (including KPs), SNs (including RIBS) and IOP solutions (device, service and information level IOP solutions) operating at different levels of M3 in NoTA based M3 systems were modelled as described in Section V.

c. Simulation parameters

The simulations were carried out in WLAN environment. The simulation parameters for physical and MAC layer are adjusted by assigning them the values shown in Table II. The parameters include the IEEE 802.11 DCF configuration parameters and the value of channel bit rate.

TABLE II. EXPERIMENT PARAMETERS

Parameters	Values
SIFS	10 micro seconds
DIFS	50 micro seconds
Slot Interval	20 micro seconds
Preamble Length	144 bits
PLCP header Length	48 bits
Channel bit rate	2 Mbps
CWmin	32
CWmax	2048
CWo	32
EW	16

VII. PERFORMANCE RESULTS

At first, when a customer enters the restaurant, his mobile device application (an information level KP and a service level AN) contacts the RIBS (an information level SIB and a service level SN) to become aware of the available services in the Smart Space. The communication takes place via SSAP API at information level and over NoTA BSD API at service level. SSAP API functions are wrappers of NoTA BSD API functions which facilitate information sharing.

After knowing about the available services, the end-user application requests the music videos on the customer's direction. The video frames are streamed form the MusicVideoClient AN to the mobile device of the Personal mobile device of a customer via NoTA BSD API functions instead of SSAP API functions. The customer invokes other ANs one by one, switching between available services after 3→5 minutes each, the ANs then invoke the corresponding SNs to provide the required services to the application.

Each AN and SN workload model is mapped to its respective platforms as shown in Figure 10. The resultant performance model is run to obtain performance results of each platform (including its hosted ANs and SNs) separately. The performance results are written to a text file in the form of different sections, one for each platform. The section of each platform contains a separate subsection for the platform component performance, M3 device level, M3 service level, and M3 information level. We only present the performance results of the platform hosting the MusicVideoServerSN. The performance results of other platforms also contain the similar information.

A. Performance Results (Platform Components)

Since the MusicVideoServerSN was implemented entirely as software, the Graphics Accelerator and Image Processor Services available from the platform were not used. Therefore, only the utilization of the processor cores of platform hosting MusicVideoServer SN is shown in Figure 12. The simulation was run for streaming of 10, 100 and 1000 packets. The solid bar corresponds to 10 packets, bar with horizontal pattern shows use-case of 100 packets and diagonal pattern correspond to 1000 packets respectively.

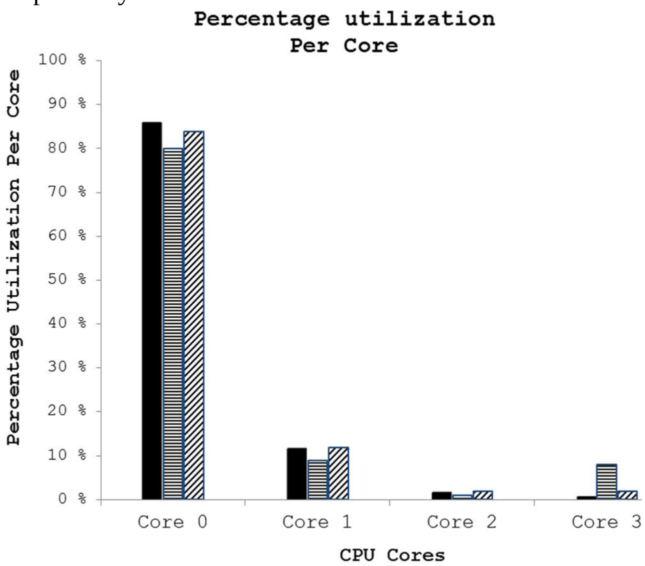

Figure 12. Utilization time of processor cores as compared to overall Utilization time of the CPU

The similar information is reported for other platform components. The cache hits, misses and accesses are also reported as described in [8] and are not shown in this case study.

B. Performance Results (M3 device level)

The performance statistics related to the M3 device level services (MAC and Transport protocols) are recorded via probes. The performance statistics of Transport (UDP) and MAC layer protocol models are shown in Table III.

TABLE III. MAC AND TRANSPORT PERFORMANCE STATISTICS

MAC/Transport Performance statistic	Values
Throughput at MAC-Level (ratio of successful Frame transmissions and total Frame transmissions)	.99
Throughput at Transport-Level (ratio of successful Packet transmissions and total Packet transmissions)	.98
Average Frame Delays	.52 millisec
Average Transport Delays	1.7 millisec
Frame loss rate(Percent)	.022%
Packet Loss rate(Percent)	.983%

The results in Table III satisfy the non-functional property (FrameRate) only if all the functions in the

MusicVideoServer SN, which make use of these OS Services, satisfy the non-functional properties from the design perspective. In case of MusicVideoServer SN, only the SendImage function makes use of NoTA API function (Hsend() for sending image data), which in turn uses the transport and MAC layer ABSOLUT protocols. As shown in Figure 8, this function must be executed within 25 milliseconds (1/40 seconds) in order to satisfy the required FrameRate of 40 Frames/Second. The processing time of this function along with the other application functions are presented in next subsection.

C. *Performance Results (Application). Validating Non-Functional Properties*

By analysing the processing times of the application source code and the percentage utilization of multi-core processor model by different external library and user-space code, we can find the potential bottlenecks in the application implementation, which will help to perform required optimizations. In other words, after identifying the functionalities which can affect a particular non-functional property, the processing times of these functionalities are analysed to find out whether the implementation of the software components satisfies this non-functional property.

We now elaborate the way non-functional property the FrameRate is analysed and validated by the performance simulation results. This non-functional property is annotated in the application syntactical view and refined to three non-functional properties in the extended behavioural view as shown in Figure 8. It is shown that due to the pipelined nature of the execution of these functionalities, each of these functionalities must be executed within 1/40 seconds (25 milliseconds) in order to achieve a frame rate of 40 frames/seconds. These functionalities and their corresponding (OpenCV library [26] functions are shown in Table IV.

TABLE IV. SHORTLISTED FUNCTIONS THAT CAN AFFECT THE FRAME RATE (A NON-FUNCTIONAL PROPERTY) OF FACETRACKERSTREAMERSERVER

Functionality	Shortlisted Function
Get a frame from Selected File	cvQuerryFrame
Create Image from Frame	cvCreateImage
Send the Image	SendImage

The processing times and the percentage processor utilization of the aforementioned functions are shown in Figure 13 and Figure 14. It is seen that all the operations are performed within 12 milliseconds. The results show that the SendImage function takes less than 6 milliseconds which is well below 25 milliseconds required to achieve the required FrameRate. In this way, the results presented in Table III are also validated. In other words, the performance of MAC and transport protocols is sufficient to satisfy the use-case.

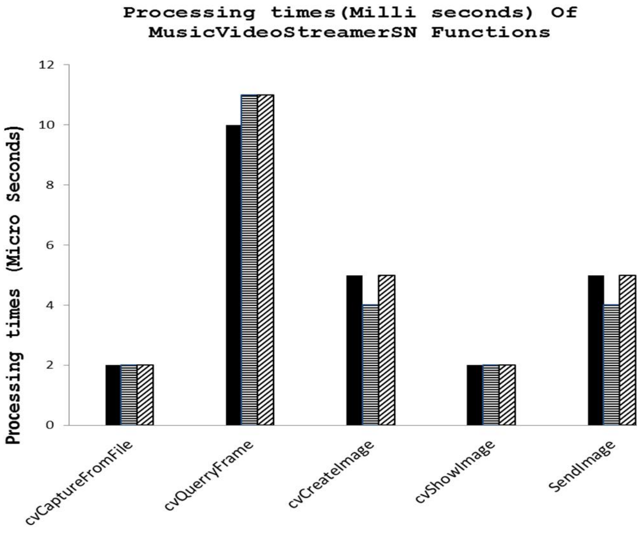

Figure 13. Execution times of functionalities attributing to Frame rate in Face Tracker Subsystem

The processor utilization graph shows that cvQuerryFrame, which fetches a frame for sending it to corresponding AN takes 54% of the overall CPU time taken by the execution of the RM application.

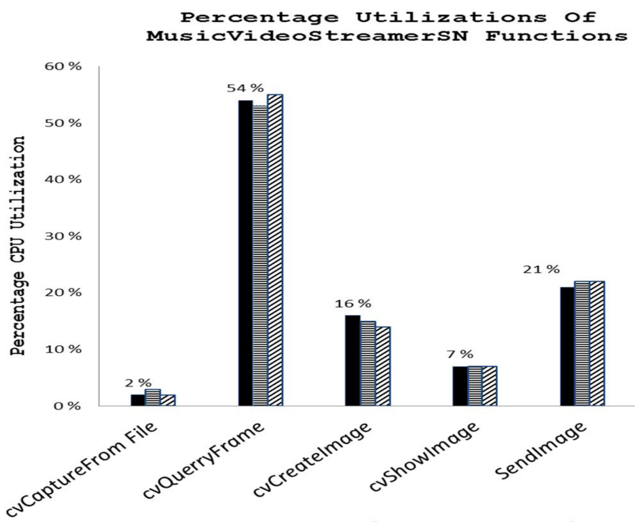

Figure 14. Execution times of functionalities attributing to Frame rate in Face Tracker Streamer Subsystem

The obtained performance results are used to perform appropriate changes in the application models by replacing the software components with more lightweight implementations or by making changes in the platform model if the performance requirements (non-functional properties) are not met. If the performance requirements are met by all the platform and software components, the architectural exploration stops and the implementation phase starts.

A. *Performance results M3 Service Level*

The processing times of a subset of the NoTA BSD API functions (used by the MusicVideoStreamerSN) are shown in Figure 15. The results show that the average processing times of all the NoTA API functions used by the SN are below 1 millisecond and therefore NoTA does not act as a performance bottleneck. Hence NoTA does not need to be replaced by a more optimized and lightweight Service-Level IOP such as ADIOS [31].

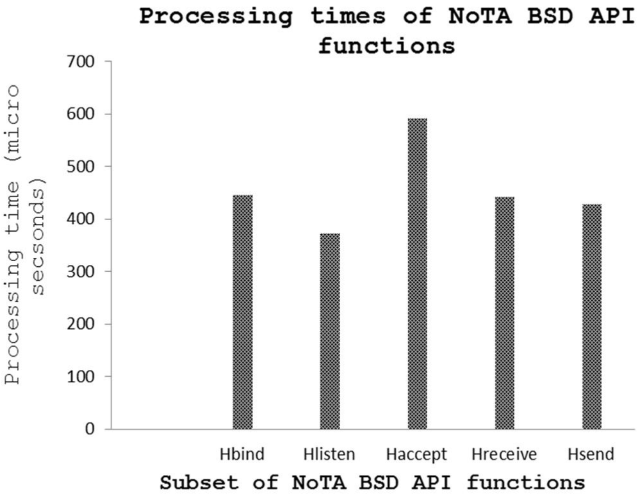

Figure 15. Processing times of M3 service level IOP (NoTA DIP).

B. Performance results M3 information Level

The processing times of a subset of the SSAP API functions are shown in Figure 16. The processing times were recorded inside the end-user application, which uses the SSAP API functions to communicate with RIBS for retrieving the information regarding the available services in the Smart Space. The results show that the processing times of all the SSAP API functions are below 1 millisecond and therefore RIBS and SSAPI do not act as a performance bottleneck.

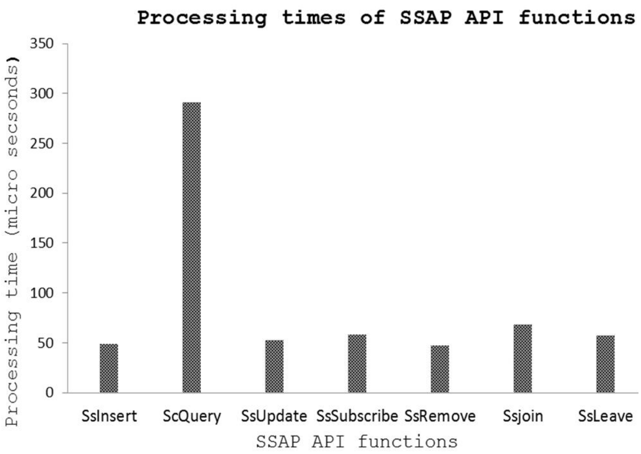

Figure 16. Processing times of SSAPI functions measured from inside user application (KP).

VIII. CONCLUSION AND FUTURE WORK

The system level performance simulation of NoTA based M3 systems was demonstrated using a Restaurant Multimedia application case study. The NoTA application previously presented in [8] was re-modelled and re-implemented at the information level of M3. The used information level entities were SSAP API, KPs and RIBS. The application design methodology describes the information level in a separate application view. UML2.0 MARTE profile and Papyrus were used as the modelling language and toolset respectively.

It was shown that the performance bottlenecks at the device, service and information level IOP solutions can be identified for performing optimizations before the development and deployment phase of M3 systems. The performance evaluation phase shows that the performance results of each M3 level, the platform components and Application can be reported and analysed separately.

The ABSOLUT methodology can be further improved by developing a GUI for easy instantiation of platform models, application workload models and selection of different protocols operating at different M3 IOP levels.

REFERENCES

[1] http://msdn.microsoft.com/en-us/library/dd129911.aspx.

[2] Aarts, E. Ambient Intelligence: a multimedia perspective. IEEE Multimedia, January-March 2004, vol. 11, 12-19.

[3] Chen, H. An Intelligent Broker Architecture for Pervasive Context-Aware Systems. Ph.D. dissertation, University of Maryland, 2004.

[4] Lappeteläinen, Antti et. al., 'Networked Systems, Services, and Information - The Ultimate Digital Convergence'. 1st International Conference on Network on Terminal Architecture, June 11, 2008, Helsinki.

[5] Kiljander, Jussi; Keinänen, Kari; Eteläperä, Matti; Takalo-Mattila, Janne; Soininen, Juha-Pekka.Autonomous file sharing for smart environments.PECCS 2011, Vilamoura, Algarve, Portugal, March 5 - 7, 2011PECCS 2011 Proceedings of the 1st International Conference on Pervasive and Embedded Computing and Communication Systems (2011), 191-196.

[6] Kienhuis, E. Deprettere, K. Vissers and P. van der Wolf. Approach for quantitative analysis of application-specific dataflow architectures. The IEEE International Conference on Application-Specific Systems, Architectures and Processors (ASAP '97), pp. 338–349, Zurich, Switzerland. July 1997.

[7] Jari Kreku, Mika Hoppari, Tuomo Kestila, Yang Qu, Juha-Pekka Soininen and Kari Tiensyrja, Combining UML2 and SystemC Application Platform Modelling for Performance Evaluation of Real-Time Systems, EURASIP Journal on Embedded Systems, volume 2008, ARTICLE ID 712329.

[8] Khan, Subayal; Saastamoinen, Jukka; Nurmi,Jari. (2011c).System-level performance evaluation of distributed multi-core NoTA systems. 2nd IEEE International Conference on Networked Embedded Systems for Enterprise Applications. NESEA 2011, Fremantle, Dec. 8-9, 2011. IEEE (2011).

[9] Tero Kangas, "Methods and Implementations for Automated System on Chip Architecture Exploration", PhD Thesis, Tampere University of Technology, Publication 616, 2006, 181 pages.

[10] USC Center for Software Engineering, Guidelines for Model-Based (System) Architecting and Software Engineering, http://sunset.usc.edu/research/MBASE, 2003.

[11] Garlan, R. T. Monroe and D. Wile. Acme: An Architecture Description Interchange Language. Proceedings of CASCON '97, November 1997.

[12] R. Roshande, B. Schmerl, N. Medvidovic and D. Garlan, D. Zhang. Understanding tradeoffs among different architectural modeling approaches. Software Architecture, 2004. WICSA 2004. Proceedings. Fourth Working IEEE/IFIP Conference on software architecture.12-15 June 2004, pp. 47 – 56.

[13] P. Lieverse, P. van der Wolf, K. Vissers, and E. Deprettere, A methodology for architecture exploration of heterogeneous signal processing systems. Kluwer Journal of VLSI Signal Processing 29 (3), 2001, pp. 197-207.

[14] Pimentel and C. Erbas. A Systematic Approach to Exploring Embedded System Architectures at Multiple Abstraction Levels. IEEE Transactions on Computers, vol. 55, no. 2, Feb. 2006, pp.99 –112.

[15] T. Wild, A. Herkersdorf and G.-Y. Lee. TAPES—Trace-based architecture performance evaluation with SystemC. Design Automation for Embedded Systems, Vol. 10, Numbers 2-3, Special Issue on SystemC-based System Modeling, Verification and Synthesis, 2006, pp 157-179.

[16] D. Manzaroli, L. Roffia, T. Salmon Cinotti, E. Ovaska, P. Azzoni, V, Nannini and S. Mattarozzi, "Smart-M3 and OSGi:

The Interoperability Platform", *In Proc. SISS 2010*, IEEE Press, 2010, pp. 1053-1058.
[17] O. Lassila, Programming Semantic Web Applications: A Synthesis of Knowledge Representation and Semi-Structured Data, doctoral dissertation, Helsinki University of Technology, Department of Computer Science and Engineering, Laboratory of Software Technology, 2007.
[18] Instantiation and feasibility evaluation of NoTA SOAD via MARTE profile and binary instrumentation. Khan, Subayal; Tiensyrjä, Kari; Nurmi, Jari. The 7th International Conference and Expo on Emerging Technologies for a Smarter World. CEWIT 2010. Incheon, 27 - 29 Sep. 2010.
[19] http://dev.iptc.org/rNews
[20] http://musicontology.com/
[21] http://www.intelleo.eu/ontologies/vw/movies/spec/
[22] Subayal Khan, Susanna Pantsar-Syväniemi, Jari Kreku, Kari Tiensyrjä and Juha-Pekka Soininen, "Linking GENESYS Application Architecture Modelling with Platform Performance Simulation", in Proceedings of the 12th Forum on Specification and Design Languages (FDL 2009), September 22-24, Sophia Antipolis, France, 2009.
[23] Multi-threading support for system-level performance simulation of multi-core architectures. ARCS 2011 - 24th International Conference on Architecture of Computing Systems.Como, Italy. 02/22/2011- 02/23/2011. Saastamoinen, Jukka; Khan, Subayal; Tiensyrjä, Kari.
[24] Application workload model generation methodologies for system-level design exploration. DASIP 2011. November 2-4 2011.Tampere, Finland. Jukka Saastamoinen, Jari Kreku.
[25] Subayal Khan, Jukka Saastamoinen, Mikko Majanen, Jyrki Huusko and Jari Nurmi. Analyzing Transport and MAC Layer in System-Level Performance Simulation, International Symposium on System-on-Chip 2011,Tampere, Finland, October 31 - November 2, 2011.
[26] http://opencv.willowgarage.com/wiki/
[27] http://projects.developer.nokia.com/NoTA
[28] Khan, Subayal; Saastamoinen, Jukka; Nurmi,Jari. (2011c).System-level performance evaluation of distributed multi-core NoTA systems. 2nd IEEE International Conference on Networked Embedded Systems for Enterprise Applications. NESEA 2011, Fremantle, Dec. 8-9, 2011. IEEE (2011).
[29] http://www.isi.edu/nsnam/ns/
[30] http://www.omnetpp.org/
[31] Eteläperä, Matti; Keinänen, Kari; Kiljander, Jussi. Feasibility Evaluation of M3 Smart Space Broker Implementations Applications and the Internet (SAINT), 2011 IEEE/IPSJ 11th International Symposium on (2011), 292-296.
[32] A UML Profile for Modeling and Analysis of Real-Time and Embedded Systems (MARTE), www.omg.org/technology/documents/profile_catalog.html
[33] Resource Description Framework, www.w3.org/RDF/
[34] RDF Semantics, www.w3.org/TR/rdf-mt/
[35] OWL 2 Web Ontology Language Reference, www.w3.org/TR/owl-ref/F