Context-aware Multimodal Feedback in a Smart Environment

Didier Perroud, Leonardo Angelini, Elena Mugellini, Omar Abou Khaled
Department of Information and Communication Technology
University of Applied Sciences of Western Switzerland
Fribourg, Switzerland
{didier.perroud, leonardo.angelini, elena.mugellini, omar.aboukhaled}@hefr.ch

Abstract - **The use of multimodality improves interaction between the user and the computer. Particularly, the use of multimodal feedback within a smart environment facilitates the integration of technology into the daily activities of the user. However the choice of the suitable output modalities requires knowledge of the user context to be effective. This paper presents an approach for the generation of context-aware multimodal feedback in the context of ambient intelligence. Our solution is based on the NAIF Framework, which handles the creation and management of a smart environment. A preliminary prototype has been developed and tested in order to validate the proposed approach.**

Keywords - multimodal feedback; multimodal fusion; context-aware; smart environment; ambient intelligence; ubiquitous computing; NAIF Framework

I. INTRODUCTION

The technological development of last years has considerably changed our daily life. Many electronic devices populate our environment and their use has become a habit. It is practically impossible to imagine a day without using a mobile phone, a computer or a television. The use of electronic devices covers most of our activities, whether related to work, entertainment or learning. The technical maturity of production means for electronic components allows devices to be more powerful, smaller in size and equipped with an increased number of functionalities. Latest devices have several on-board sensors that allow improvements of the usability. Thanks to them the local context is taken into account during software development. For example, a smart phone can now detect its orientation to adapt the graphical representation of an application [1], can take into account light condition to change the luminosity of the screen and use head proximity sensors to turn off the display when answering a call.

Miniaturization and reduced costs of electronic components encourages manufacturers to integrate multiple communications interfaces in devices. This integration extends the functional capabilities of each device. For example, nowadays it is possible to find on the market televisions that allow direct access to Internet. The widespread diffusion of communication means and the ubiquitous sensors integration open the door to the exchange of information between devices in a given environment. The increased connectivity and information exchange between systems are the basis for developing novel, intelligent applications. The primary purpose of these intelligent applications is to provide a greater control of the environment to the people. This field is called *Ambient Intelligence*, as described by P. Remagnino et al. [2], with some typical scenarios presented in [3].

The daily presence of people in an ambient intelligent environment involves a change of habit in terms of interaction. The smart environment must be non-intrusive and able to understand user's needs. The distribution of systems in the space also requires the use of another way of interaction than standard mouse and keyboard. Indeed, the user must be able to interact with its environment without constraints on the devices to use. Multimodality seems to be a suitable and flexible solution for the interaction between the user and a smart environment.

The work presented in this paper is related to multimodal generation of output content, taking into account user context in an intelligent environment composed of autonomous distributed systems. Issues concerning the creation of an intelligent environment will not be discussed in this paper. The approach proposed for the multimodal output generation uses NAIF (Natural Ambient Intelligence Framework) [13], which allows the setup of an intelligent environment.

The paper is organized as following. Section II presents some work in the scientific community addressing the concepts of multimodal fission and multimodal generation with context management. NAIF Framework is briefly presented in Section III. The proposed approach to context-aware multimodal feedback generation is explained in Section IV. Section V presents a prototype that validates the use of the proposed approach. Section VI concludes the paper and discusses future work.

II. RELATED WORK

The use of multimodality in human-computer interaction is a subject frequently discussed in the scientific community. The acquisition of multiple signals from the human is the first big challenge to solve in order to build systems more comfortable for the user. Nevertheless, the output generation should be also investigated to grant multimodality in both directions of the human-computer interaction. Few projects have dealt with multimodality in both input and output processes, the most related to our work are presented in the following paragraph. To the best of our knowledge no one treated input and output

 ISBN: 978-1-61208-170-0

multimodality using context information from a smart environment.

SmartKom [4] is a multimodal communication system that combines voice, gesture and facial expression as input and output. The aim of this project is to create a natural experiment of communication between user and machine. The concept is based on a virtual agent capable of interpreting communicative intentions in the context of assistance to purchase a ticket. This project is relatively complex since it deals with multimodality in a full and symmetrical spectrum. The generation of multimodal output of SmartKom is based on a system developed under another project called WIP [5][6]. The main component of WIP is a presentation planner that transforms the intentions of communication in presentation tasks. Then the planner allocates these tasks to specific generators of modality like voice or gesture. The fission engine of SmartKom uses therefore an important knowledge database that contains all different patterns and presentation strategies available for each modality. SmartKom is an example of project that addresses the challenges of multimodality as input and output. Its development is centered on the user and multimodality is limited to a few specific modalities. The system SmartKom cannot be applied in a distributed environment like a smart environment.

C. Rousseau et al. [7] proposed a conceptual model called WWHT for the multimodal presentation of information. The model WWHT is articulated around four basic concepts, *What*, *Which*, *How* and *Then,* describing the life cycle of a multimodal presentation adapted to the context of ongoing interaction. The presentation process is based on 4 components: the information to present, the interaction components of the system, the ongoing context of the interaction and the resulting multimodal presentation. A first semantic fission of information occurs in the step *What* in order to form smaller units of information. In the next step, *Which,* the various units of information are allocated to a specific modality (e.g., visual or haptic) and the choice of communication medium is made (screen, sound speaker). During this step, the dependencies between allocated modalities are also assessed. Multimodality is addressed in this assessment by considering the CASE model [8], which classifies the possible combinations of modalities. The generation of modalities occurs in step *How* where the interaction components of the system produce outputs. Finally the step *Then* is responsible of context monitoring in order to adapt multimodal presentation.

POPEL [9][10] is a generating component of natural language integrated in the XTRA Framework. XTRA offers a treatment of multimodality as input and output. Supported communication channels are focused on natural language and complementarity by gestures. Like WWHT, Popel separates the information fission process of output generation. The first step *POPEL-WHAT* aims at the selection of information to be transmitted depending on the current context. *POPEL-HOW* is responsible for generating output. WWHT and POPEL use a concept of division between the modality and its representation. This separation makes the model of WWHT or the implementation of POPEL much more flexible. It also helps break down the complexity of multimodality processing.

W3C Multimodal Interaction Framework [11] is a specification provided by W3C to extend the Web for supporting several modes of interaction. The specification addresses the multimodal interaction as input and output. It describes the different components that any multimodal system must implement. The internal operations of components are not included in the description. The W3C specification separates the generation from the rendering of multimodality. The exchange of information between components is also specified. It consists of several markup languages using the eXtensible Markup Language (XML) specification of W3C. For example, the component audio rendering can handle a Speech Synthesis Markup Language (SSML) document while a graphics rendering component can interpret an eXtensible HyperText Markup Language (XHTML). The use of a standardized language to exchange information improves the modularity of the framework. This concept is particularly suitable in a distributed environment context.

The DynAMITE project exploited multimodality in a smart environment where heterogeneous devices can interoperate thanks to a common framework. This work deals with the ubiquitous computing within dynamic ad-hoc devices ensembles. Even if multimodality is addressed at both input and output sides, the context information of the environment is not exploited in this project. Unfortunately the project DynAMITE seems to have been abandoned. Some explanations about the project in general or about the internal components can be found in [14][15][16].

Our approach presented in this paper focuses on the context-aware multimodal generation within an intelligent environment. The execution context of the applications is a distributed environment, where multiple applications can run in parallel. Therefore, our approach must be as flexible as possible because each application requires different needs in terms of generation of the multimodal feedback. Our approach uses some concepts of the aforementioned works that improve flexibility and modularity. We separate the notion of modality and representation. We also use markup language to exchange information within the environment. Moreover our framework aims to address the issue of a context-aware multimodal feedback on multiple distributed systems in a smart environment.

III. NAIF FRAMEWORK

The generation of multimodal output presented in this paper takes place in a smart environment. Our approach is based on NAIF [12], which is an acronym for *Natural Ambient Intelligent Framework*; it is developed since 2009 at the University of Applied Sciences of Western Switzerland, in Fribourg. This framework aims to address the issues related to the setup and development of a smart environment. This section is a brief presentation of the framework; further details are available in [13].

 ISBN: 978-1-61208-170-0

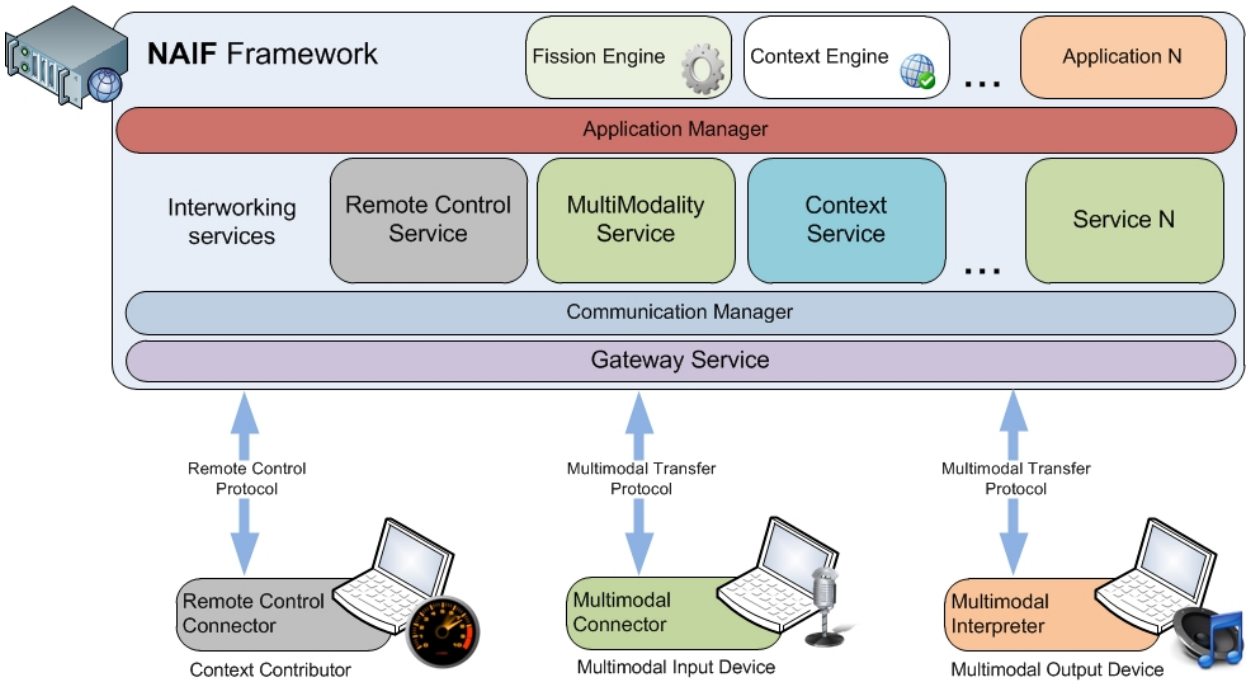

Figure 1: Context-aware multimodal generation architecture based on NAIF

The basic idea of NAIF is that our everyday environment is full of devices performing specific tasks independently. Each of these devices has inherent capabilities that can be shared with other systems of the environment. Therefore, the devices can benefit from the presence of other systems to do their own tasks or accomplish a common goal. For example, a smart phone could use a radio in another room to broadcast the ringing of a phone call when the user is not present. In the same way, the smart phone could collaborate with an application of energy saving by sharing onboard light sensor measures to signal light on.

NAIF is built on client-server architecture. A central platform provides services to devices that constitutes smart environment. The operation of NAIF is based on three concepts described below.

1) Intercommunication

Each system of the smart environment hosts a *software agent*. This agent manages the communication with the central platform and the data exchange format. On the server side, a service called *Gateway* is responsible for routing communication between systems. To support heterogeneity, the communication protocol consists of XML frames. The protocol NAIF is therefore located at application level and relies on a TCP stack.

2) Interworking services

To ensure interoperability between systems in the environment, the central platform offers on-demand services. These services are mandated to make collaboration between systems effective but also to facilitate the work of developers that want to exploit the advantages offered by the available systems. For example, NAIF provides a service called *Remote Control Service* that shares measures of sensors available on the devices of the environment. The service layer of the platform is extensible. Developers can add services as needed.

3) Multimodal interaction

NAIF proposes an approach to multimodal interaction in smart environments. An interworking service called *MultiModality Service* is provided by the central platform. This service operates as a directory that allows sharing modalities of systems equipped with appropriate devices and software, which is charged to process signals from onboard sensors. For example, a system with a speech recognition engine and a microphone can share the detection of words pronounced by user. This service deals with modalities only as input and has no mechanism of fusion.

The following section presents our context-aware multimodal feedback concept integrated in NAIF.

IV. CONTEXT-AWARE MULTIMODAL GENERATION

As discussed above, the presence of a user in a smart environment involves new ways of interaction. This problem is often approached from the point of view of combining and exploiting different input interaction modalities. The concept presented in this paper focuses on combining and exploiting different *output* interaction modalities. As the user benefits from multimodality as a natural channel to communicate with the environment, the environment must be able to produce a multimodal feedback to the user. The generation of feedback should occur under the best conditions possible to improve comfort for the user. To find the best conditions, the user context should be considered. Context data will allow the smart environment to produce an optimal multimodal feedback. For example, if the user has a visual impairment, an auditory feedback should be produced instead of a visual one.

 ISBN: 978-1-61208-170-0

The concept of context-aware multimodal generation explained in this section is an extension of NAIF. The extensibility of the framework allows the inclusion of new interworking services. The solution detailed in this paper is therefore based on the integration of new services and software components. Figure 1 shows the architecture of NAIF with extensions. The following sections describe this architecture extension.

A. Multimodal Input Sharing

As previously explained, NAIF includes a *MultiModality Service*, which allows the sharing of input modalities. The idea behind this concept is based on the collaboration between systems in a smart environment. Imagine a TV, it requires a remote control to understand user commands. If a system in the environment has a voice recognition engine, the TV could benefit from it to improve its interaction input channels. The current version of the *MultiModality Service* operates as a directory. Systems capable of processing input signals from users publish their functionalities in the directory. Systems wishing to receive a modality may register to the directory. When a modality is detected, e.g., a word is spoken, a notification will be sent to all systems that observe it.

The concept of modality is very subjective. For example, it is difficult to characterize what a voice modality is: it can contain just a word, a phrase, an intonation or a pronunciation. NAIF avoids this problem by allowing developers of different applications to model their own modality according to their needs. Each modality will be described in an XML schema. Schemas will be shared with all systems of the smart environment using the *MultiModality Service* through a software component called *Multimodal Connector*. The *Multimodal Connector* component is hosted on the environment systems that use the *MultiModality Service*. The *Multimodal Connector* is responsible of managing the link between the system and the interworking service. It uses the *Multimodal Transfer Protocol* to communicate with the service. This XML protocol is encapsulated within the data field of NAIF frames. The structure of this protocol is simple. It contains only the transfer of commands like *publish* a modality, *register* to a modality or *notify* a modality. As the schema of each modality is available in the connector, systems can interpret the contents of the frames received.

B. Multimodal Output Generation

The generation of multimodal feedback in our framework is designed in order to allow a system that has limited feedback capabilities to use the outputs of other systems. A heating controller for example will be able to use a TV screen to display an alert when resources are missing. The designed approach extends the possibilities of the *MultiModality Service*. By adding a new *publish* command, a system can offer its feedback capabilities to other systems. The service can now receive two types of commands. The first announces a system as a supplier of modality, the second as a provider of feedback. This extension requires a change in the *Multimodal Connector* and a protocol arrangement. A generate-modality message to feedback generator systems is added in the *Multimodal Transfer Protocol*. This change raises a new problem. Generating systems have to interpret this modality message to produce feedback. To solve this problem, a new software component called *Multimodal Interpreter* is placed on the published multimodal feedback generator systems.

The multimodal interpreter is responsible for the transformation of a modality in an effective presentation of the information. This component depends on the local platform. It cannot be totally generic. The concept of modality in the *MultiModality Service* is not limited for input; the developer can specify its own modality for output as well, if needed. The interpreter must however be able to understand modalities that are sent for generation. In summary, if a system sends text as voice modality, the work of the *Multimodal Interpreter* is to synthesize the voice through text-to-speech.

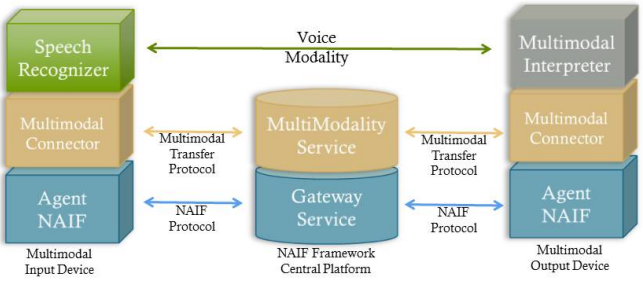

Figure 2: Software stack of an exchange of voice modality between a speech recognizer system as input and a published multimodal output system.

The proposed extension involves the creation of roles as shown in the architecture in Figure 1. A system can act as a multimodal input device. In this case it only hosts the *Multimodal Connector* and shares its modalities. A system can also act as multimodal output device. In addition to the connector, the output system hosts the *Multimodal Interpreter* as shown in Figure 2 and generates received modalities. Input and output roles can be combined.

C. Context management

In our work, we have defined the context as the description of the parameters surrounding an action. As part of a smart environment, the context concerns users as well as all the environmental parameters that affect a given action. In terms of multimodal feedback generation, the context is a primordial factor that can disrupt interaction with the user. A context management should be present to produce a good quality of feedbacks.

Our approach is based on the integration of a contextual reasoning engine and an interworking service in NAIF. The *Context Engine* is located at the application layer of the framework. This layer allows applications to run on the central platform and to access interworking services through an *Application Manager*. The role of *Context Engine* is to extract the context state from available data of the smart environment. The management of the context adds a new role for the systems of the environment. Systems that act as *Context Contributor* publish data from their sensors, or from their local context. A smart phone for example can publish its current direction sensor value or its current activity. The publication of data sensors is available in NAIF through the *Remote Control Service*. The devices can use the *Remote Control Connector* that links to the service on the central platform. Therefore the

 ISBN: 978-1-61208-170-0

context engine can use sensor data available in the *Remote Control Service* to extract a global context. The engine then builds a representation of the context that is stored in the new service called *Context Service*. This service operates as a repository of the current state of the smart environment. Each system of the smart environment can access this service to obtain information about the ambient context. This allows preserving the autonomy of the different systems within the environment.

A first version of the *Context Service* has been developed. However, at the moment, only the software components running on the central platform can access the *Context Service* (the other systems in the environment cannot access this service because no connector component exists). An effective representation of a context must also be studied. This modeling process is also a subjective task, which could limit the functionality of the framework. One might ask if the context and the reasoning engine should not be a single entity. To maintain the flexibility of NAIF, the *Context Service* has been separated from the context-reasoning engine. This allows to easily change the engine with another that implements a different algorithm.

D. Multimodal fission

The generation of multimodal feedback is effective through the *MultiModality Service*, thus the framework should be able to split information into multiple modalities. Fission can be approached from two aspects. The first approach is the fission of information semantics in order to extract multiple information units, which will be processed into modalities. This task of division is extremely complex and requires an important research work. Instead the concept of fission addressed in our approach concerns the choice of the best modalities for a specific feedback that has to be sent to the user.

We introduced a component called *Fission Engine* in the application layer of the central platform. The engine receives feedback intentions as input, and then takes care of electing the best modalities and the best systems of the environment to generate these feedbacks. The Fission Engine relies on context state to select modalities and systems. It consults the *Context Service* directly to obtain the current context state. Insofar as the choice requires additional reasoning, we can imagine that the *Context Engine* offers facilities to solve the problem of election. The *Fission Engine* communicates directly with the *MultiModality Service*. Upon receiving a feedback intent, it chooses the best modalities and then it sends a feedback generation message to the selected systems. Take the example of a security application that controls the entrance of a room. When an alert message should be sent, the application formulates its intent to the *Fission Engine*, which will then select the best modalities on the best feedback generator systems. Taking into account the context of the user's proximity, for example, the *Fission Engine* could ask the generation of visual and audio alert on the systems closest to the user.

Like the reasoning engine of the context, there are many possible approaches to implement the *Fission Engine*. The intelligence of the engine can fundamentally modify the effectiveness of feedback. The management of multimodality is also reflected in the engine. The CASE properties [8] can be used in the election procedure to improve feedback. The *Fission Engine* is positioned in the application layer on the central platform. Therefore this software component can be exchanged easily and the flexibility of NAIF is preserved.

V. PROTOTYPE

A prototype has been developed in order to demonstrate the feasibility of the concept. The objective of the prototype is to show that it is possible to exploit the capabilities in terms of modality generation of different autonomous systems in a smart environment, taking into account the state of the ambient context. The prototype must test the different software components presented in the previous sections. The architecture of this prototype is shown in Figure 3.

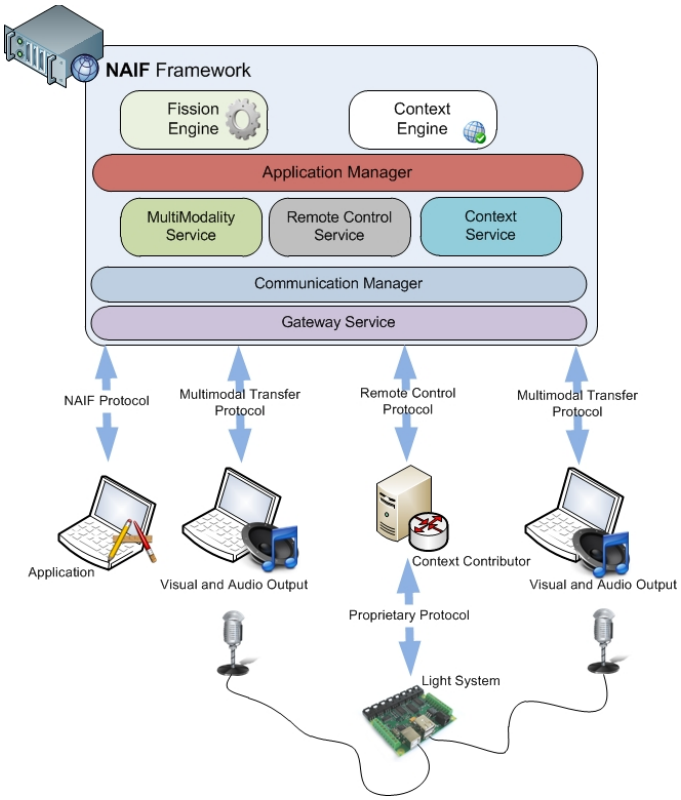

Figure 3: Prototype architecture

Two laptop placed in the environment are able to produce auditory (text-to-speech) and visual feedback. Both systems publish their generating capacity of two modalities in the *MultiModality Service*. Two sensors of noise intensity are located near these multimodal output devices. A computer processes the signals from these sensors and publishes values in the *Remote Control Service*. A *Context Engine* accesses data from *Remote Control Service* and built a simple XML representation (e.g. *<Noise location ='1'>57</Noise>*) of the sound context that it stores in the *Context Service*. By subscribing to notifications of the *Remote Control Service,* the *Context Engine* performs regular updates of the context representation.

An application running on a third laptop with no audio output requires the display of a text and a vocal commentary as

 ISBN: 978-1-61208-170-0

feedback. Using its NAIF communication agent, it sends feedback intent to the *Fission Engine*. Its frame is structured as follows:

<Feedback><Text>textual feedback</Text>
<Voice>vocal feedback</Voice></Feedback>

The *Fission Engine* queries the *MultiModality Service* to find available output generators and it analyzes the XML representation in the *Context Service.* Then, it chooses the modalities and selects the best feedback generators according to the noise state of the context. Finally the *Fission Engine* sends modality generation requests to the selected systems through the *MultiModality Service*.

Feedback generation requests are received in the *Multimodal Interpreter* of the selected output devices. Modalities are ultimately generated by the respective hardware. In this prototype, only the sound context is taken into account. The intelligence of *Fission Engine* is very simple too. Nevertheless, the voice generation and display of text are produced on the system the least disturbed by noise.

The results obtained are consistent with the objectives. The integration of our concept in NAIF let us take advantage of the flexibility of the Framework. The application could for example make the fission itself and directly contact *MultiModality Service* to send a generation request using the *Multimodal Connector*, or it could use the *Fission Engine* as explained in this prototype.

VI. CONCLUSION AND FUTURE WORKS

In this paper, we proposed a context-aware approach to multimodal feedback generation in a smart environment. The context analysis is included in the solution to improve the efficiency of multimodal generation. Our proposal is designed as an extension of the NAIF Framework, which allows setting up, and developing an intelligent environment.

The contribution of our approach could help the development of intelligent environments. By using multimodal feedback, ambient intelligence can communicate with the user trough more natural channels. Therefore, the integration of smart environments in everyday human life could be less complicated. Based on the context, ambient intelligence can also react to disturbances that might limit the usability of the global system. Our contribution relative to the NAIF Framework provides new features for developers of applications in smart environments. It is now possible to benefit from the intrinsic characteristics of autonomous systems in terms of output. The developer can also focus on the user when designing new applications because of the full spectrum of the multimodal communication with the user. Our extension therefore complies with the conception of the Framework.

The prototype presented in this paper validates the proposed context-aware fission concept, however some improvements are still needed. The reasoning engine of the context should be improved with algorithms and technologies that will be used to build the representation of the context. An approach based on ontologies, for example, could meet the needs of our concept.

Finally, the use of the *Fission Engine* must be improved, mainly in the formulation of feedback intentions. This concept should be modeled to establish the communication protocol used to contact the *Fission Engine*. The concepts of modalities and feedbacks should be further investigated

VII. REFERENCES

[1] S. Valbert and C. Per Thorsø, " Improving usability of mobile devices by means of accelerometers ", Master Thesis, Kongens Lyngby, 2009, ISBN: IMM-M.Sc.-2009-32.

[2] P. Remagnino and G.L. Foresti, " Ambient intelligence: a new multidisciplinary paradigm ", IEEE Xplore, vol. 35, 2005, pp. 1-6, doi: 10.1109/TSMCA.2004.838456 .

[3] K. Ducatel, M. Bogdanowicz, F. Scapolo, J. Leijten, and J-C. Burgelman, " Scenarios for ambient intelligence in 2010 ", ISTAG report, 2011, p. 54.

[4] W. Wahlster, " SmartKom : fusion and fission of speech, gestures and facial expressions", Proc. 1st International Workshop on Man-Machine Symbiotic Systems, Kyoto, Japan, 2002, pp. 213-225.

[5] W. Wahlster, E. André, W. Finkler, H.-J. Profitlich, and T. Rist, " Plan-based integration of natural language and graphics generation ", Artificial Intelligence 63, 1993, pp. 387-427, doi: 10.1016/0004-3702(93)90022-4

[6] E. André, W. Finkler, W. Graf, T. Rist, A. Schauder, and W. Wahlster, " WIP: the automatic synthesis of multimodal presentations ", Intelligent Multimedia Interfaces, AAAI Press, Menlo Park, 1993, pp. 75-93, ISBN: 0-262-63150-4

[7] C. Rousseau, Y. Bellik, and F. Vernier, "WWHT: un modèle conceptuel pour la présentation multimodale d'information", Proc. IHM, 2005, pp. 59-66, doi: 10.1145/1148550.1148558.

[8] L. Nigay and J. Coutaz, " Design space for multimodal systems: concurrent processing and data fusion", Proc. Conference on Human Factors in Computing Systems (INTERACT '93 and CHI '93), ACM, 1993, New York, pp. 172-178, doi: 10.1145/169059.169143

[9] D. Schmauks and N. Reithinger, " Generating multimodal output: conditions, advantages and problems ", Proc. 12th conference on Computational linguistics (COLING 88), Budapest, Hungry, 1988, pp. 584-588, doi: 10.3115/991719.991758

[10] N. Reithinger, " POPEL—a parallel and incremental natural language generation system ", In C. Paris, W. Swartout and W. Mann, ed, Natural Language Generation in Artificial Intelligence and Computational Linguistics, Kluwer, Dordrecht, Netherlands, 1991.

[11] http://www.w3.org/TR/mmi-framework/ 01.08.2011

[12] http://www.naif-project.ch 01.08.2011

[13] D. Perroud, F. Barras, S. Pierroz, E. Mugellini, and O. Abou Khaled, " Framework for development of a smart environment, conception and use of the naif framework ", Proc. 11th International Conference on New Technologies of Distributed Systems (NOTERE 11), Paris, France, 2011, pp 151-157.

[14] http://www-past.igd.fraunhofer.de/igd-a1/projects/dynamite/ 01.08.2011

[15] K. Richter and M. Hellenschmidt, " Interacting with the ambiance: multimodal interaction and ambient intelligence ", Architecture, vol. 19, 2004, p. 20, 2004.

[16] M. Hellenschmidt and T. Kirste, " SODAPOP: a software infrastructure supporting self-organization in intelligent environments ", Proc. Industrial Informatics (INDIN04), Berlin, 2004, pp. 479 – 486, doi: 10.1109/INDIN.2004.1417391

 ISBN: 978-1-61208-170-0