

Composite Context Information Design and Model Approach for Adaptive Service Decision

Sukyong Kim*, Eungha Kim*, Youngil Choi*

*Service Cognitive Control Team, ETRI, 138 Gajeongno, Yuseong-gu, Daejeon, Korea

{ksk0314, ehkim, yichoi}@etri.re.kr.edu

Abstract—Context-aware computing has been considered as promising topic in pervasive computing area, but a review of the existing approaches shown that developing diverse application services in this area is still motivated based on the analysis of existing approaches in literature. The purpose of this study, is proposed the composite context information model to provide the knowledge that is determine the path of adaptive service configuration in mobile environment. First of all, our research is defined the composite context information in perspective of this study, and based on this definition, is proposed the approach of master data modeling to configure the composite context information. It is show to can resolved the problem related with data processing, among a lot of difficulties that the proposed context information based system is shown.

Keywords—Composite Context Information, Context-aware

I. INTRODUCTION

Changes in the web 2.0 environment and the rapid development of mobile devices such as smart phone, to provide user-centric service, became more important to apply with the variety context information. In order to provide user-centric service, depending on the service needs, many development companies are developing their own applications by applying the open source. These developments situation, if user's needs and context to determines the services diverse, or complex, difficult to ensure the performance of the developed applications, enhanced the code complexity of those applications, since independently created code, difficult to share and reuse of the service applications regard with similar context.

In addition, when many companies to develop applications using Open API or mash-up service, if the distributed Open API services was interrupted by unsuspected situation, it can be disable since take emergency steps to deal with that problem. The occurrence of these problems, because there was a reflection of the information about the network service

In many researches related with context-awareness, suggested many approaches to manage the context information in order to decide about occurred situation. Especially, the used approach to represent about occurred context information, can be classified into technologies those are low-level context representation based on ontology, it is logic-based inference method to use first order logic and description logic, and it is high-level context representation to

use low-level context representation and rule. thus, in spite of in many ways to express the context information presented and builded many context information, it is very rare the case to reusing and sharing the context information for the construction of the system or application is needed apply the context information.. Dey [5] defined context as “any information that can be used to characterize the situation of an entity. 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 application themselves”. Also, Dey provided the following definition: “A system that uses context to provide relevant information and/or services to the user, where relevancy depends on the user’s task” for context-aware computing. Context-aware computing means that it enables an application especially for mobile to utilize knowledge about various context dimensions, such as who the user is, what the user is doing, where the user is and what computing equipment the user is using [6]. Then the application can provide specific information and customized services to users according to the interpreted context.

Pashtan[7] classify the context dimensions into four sub dimensions which is user static context, user dynamic context, network connectivity and environmental context. Each context dimension is described by its respective context parameters, for example, a user’s static context parameters are his profile, preferences, interests, and etc. shown in Table 1.

TABLE 1. CLASSIFY CONTEXT-INFORMATION

Category	Type	Remark
Physical Information	Gain from physical sensor, etc. Location, Temperature, Time	Need to classify to physical category
Virtual Context Information	status information as like Battery, memory information represent to SW or application	According to the research, classify the logical information.
Explicit input context information	Explicit input behavior by user, as like profile, place. Etc.	According to the application or system, this information is various
User client generate information	Activity	Activity information is need to abstract process

Regarding those contexts, three kinds of activity may be performed. Context gathering is conducted to collect raw

context data from various sensors and augment these data. Context interpretation consists of transforming raw sensor data into human understandable high-level contexts. High-level contexts are composed of data from different context data sources or of different context dimensions (location, temperature, etc). Interpretation is carried out by using predefined rules as in [8]. A slightly different approach is presented in [9], where context is modeled and interpreted based on ontologies. Following this approach, context is represented as predicates written in OWL and context interpretation is performed by a context reasoning engine that supports RDF-S and OWL reasoning and general rule based reasoning.

II. RELATED WORK

CoBrA is a context-management framework that was developed in 2004 by Dr. Harry Chen as part of his PhD research at the University of Maryland, Baltimore County[10]. CoBrA is the agent based architecture for supporting context-aware computing in intelligent spaces. Intelligent spaces are physical spaces populated with intelligent systems that provide pervasive computing services to users[10]. CoBrA uses OWL (Web Ontology Language) to define its ontologies for representing context and modeling. The main ontology associated with CoBrA is SOUPA (Standard Ontology for Ubiquitous and Pervasive Applications).

Dey et al.[14] introduce a conceptual framework to assist in the design of context-aware applications present a toolkit that can facilitate context-aware computing research. by allowing the empirical investigation of the design space and the exploration of difficult challenges in the handling of implicitly sensed context. In this research, the Conference Assistant which helps conference attendees to determine what presentations or demonstrations are interesting or relevant is presented.

The Context Toolkit [15] is an architecture developed at the Georgia Institute of Technology that aims to provide reusable solutions to the problems of developing context-aware applications. The Context Toolkit provides useful domain-specific abstractions for the incorporation of context data generated from sensors into applications, through the use of the widget abstraction. This based on the GUI concept of a widget as a reusable component for abstracting away from and hiding the specifics of a physical device. In Davy et al[16], a component-based approach is presented , which not only support context-aware adaptation of services, but also support adaptation of the context management system itself at deployment time and at runtime.

Baldauf[13] provides a comprehensive review and analysis of current context-aware system regarding architecture, context model, and frameworks. In this research, it is argued that developing flexible and useable context ontologies that cover the wide range of possible contexts is a challenging task. In this review, Baldauf summarizes and presents other research results about context management models, sensors, context models.

Dey[5] identified a design process for building context aware applications. There are not many researches which address a detailed and concrete design process. Karen[20] also outline the process that is generally followed when building a context-aware application. In this research, the design and implementation steps rely on the branching, triggering and context query APIs to incorporate context-aware functionality, but otherwise adopt traditional methodologies and languages.

Referring to research of Anagnostopoulos et al. [18], Jacob[17] summarize the concept of situation-awareness as “the particular kind of context-awareness, where situations are viewed as logically aggregated pieces of context”. Their definition is one of only few definitions differentiating between context and situations, but finally implies the aggregation of context only. Also Jacob[17] summarize that a situation-aware application has “to estimate the user’s current situation(s) and react appropriately” and to “autonomously adapt to the user’s current situational context”. Theodore et al. described that modeling context for representing, manipulating and accessing both static and dynamic information is essential for context-aware infrastructures [19].

Also it is shown that The significance of a flexible and practical context model is evident not only in facilitating the task of programming a context-aware system, but also in permitting the desirable degree of gathering, management, dissemination and reasoning on contextual information. After reviewing and analyzing the related researches, we found and categorized some aspects to be considered in the development for context aware system. The first is that detailed process for developing this system was not shown. The context-aware services must be able to understand various aspects of context-aware computing technology current situation or context and use them to interact with the user in a more intelligent way by combination of different technologies such as sensing devices, intelligent software, wireless technologies, etc. so that a software engineering methodology is needed to integrate the diverse nature of context aware technology[6]. Only a few researches[5,19] present a design process or development process. In one research, the design process is as follows [5]:

Step1) Specification: Specify the problem being addressed and a high-level solution.

1.1. Specify the context-aware behaviors to implement.

1.2. Determine what context is required for these behaviors (with knowledge of what is available from the environment) and request it.

Step 2) Acquisition: Determine what hardware or sensors are available to provide that context and install them.

Step 3) Action: Choose and perform context-aware behavior.

In other software engineering process suggested in [16], the steps can be partitioned into the following tasks: analysis (A), design (D), implementation or programming (P), infrastructure customization (I) and testing (T).

Step 1) The analysis task captures and documents the functionality and requirements of the application, as in most other software lifecycle models.

Step 2) software engineering process includes two additional steps specific to context-aware applications. The first focuses on the types of context information that are required in order to implement the functionality identified at step 1.

Step 3) the analyst identifies those choices and events that are influenced or triggered by the context, and, for each of these, documents the role of context by producing sample preferences and triggers.

Step 4) the software engineering process diverges into two sets of tasks that can be performed in parallel, one concerned with design and implementation, and the other with customization of the software infrastructure.

Neither of two methods did suggest detailed procedure, and then we require a development method considering ontology development and hardware technology.

The second is what characteristics which context aware application should have are and how and where those ones are reflected. In the research[14], a few reasons why handling context is difficult are described as follows, but it's not comprehensive and complete: (1) there are no guiding principles to support good software engineering practices; (2) designers lack abstractions to think about context; and (3) context sensing is very often distributed and leads to complex distributed designs. One of the possible solution for (1) is to use a layered architecture reflecting the concept of "separation of concerns" so that applications can use contextual information without worrying about the details of a sensor and how to acquire context from it. Regarding the other solution for (2) and (3), it is not clear that how they are reflected in what design process.

The third is that reusability including context ontology reuse, framework or service scenario reuse has not been considered in the development of context aware application. The cost and time can be huge if all parts of such application are developed from scratch, so we need a method to enhance reusability

III. COMPOSITE CONTEXT INFORMATION MODELING

For our study, we present composite context information concept as shown in Figure 1. First of all, we have to define the composite context information's concept for our research. We define the composite context information that is "enhanced high level context information by integration or composite context information related with multi-entities for decide or supply the service of system".

In order to provide composite contextual service, location, user and device ontology is constructed and also context aware module is built. Service history is stored into the service history repository. More detailed information is discussed in experiment result section of this paper. This composite context information is consist of unit context information belong to each entity related with various domain.

Our case study is for intelligent meeting assistant who helps a manger to organize team meeting including arrangement of meeting place, notification of meeting, operation of hardware devices. User A as team manager is scheduled to have a team

meeting but meeting place is not arranged. This scenario is described in Table 2.

For the provision of contextual services, location and device ontology are modeled and user ontology is explained in more detail.

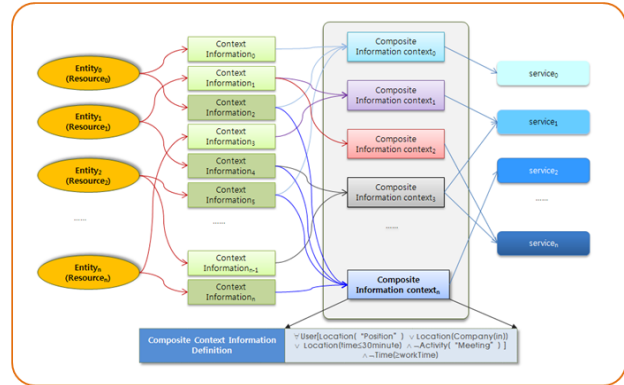


Figure 1. Composite context information concept

TABLE 2. SCENARIO FOR USER CONTEXT

User's Process and Context to related service		
Process	Context	service
Inquiry on user A schedule	After 30minutes from now, there is a team meeting but meeting place is not reserved.	Reservation service for arranging meeting room
Search of available room in the company building	A meeting room appropriate for beginning and the expected meeting time is found	Reservation of meeting room
Inquiry on user A profile	Inference of team member of user A	level-2 heading, level-3 heading, author affiliation
Temperature Detection of the meeting room	Inference of temperature preference	Cooling or heating meeting room
Beginning of meeting	Change of the status of meeting room	Change the status of meeting room to 'in use'

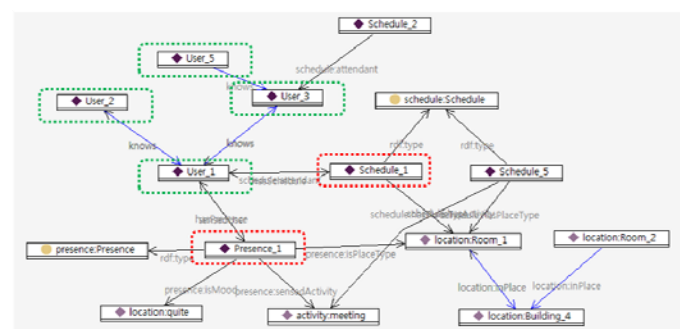
A. User Ontology

A user is assumed to have the following properties and this user class plays the most centric role in the service provision based on context awareness.

- 1) User is person and have mobile device.
- 2) User has attributes including name, job, sex, team, age etc. in user profile.
- 3) User has preference and interest.
- 4) User has a schedule

Figure 2 shows the class and attribute connected by the inference rules. The user is connected with a device and a service through "hasPreference". The presence information includes a service history

To support our scenario, if there is a meeting in User_1 schedule, the inference result shown in Figure 3 shows the arrangement of meeting Room_1 and notification of the team meeting to members. This notification is achieved through the

$$\begin{aligned} & (?x \text{ hasPresence } ?y) \wedge (?y \text{ presence:isPlaceType } ?k) \\ & \wedge (?k \text{ location:inPlace } ?p) \wedge (?x \text{ knows } ?u) \wedge \\ & \wedge (?u \text{ hasPresence } ?y1) \wedge (?y1 \text{ presence:isPlaceType } ?z1) \wedge (?z1 \text{ location:inPlace } ?p1) \wedge \\ & \text{notEqual}(?x, ?u) \wedge \text{equal}(?p, ?p1) \end{aligned}$$


B. Presence concept to represent composite context-aware



As results of applying this, context information was created, stored and interpreted to provide reactive services. Also, it showed possibility for providing proactive service based on the service prediction. Despite this experiment, we have not addressed the provision of step-by-step procedure for developing context aware application. Instead, we applied a generic software engineering method and ontological method to this application. During components of our prototype framework, components and service scenario reusability was partially considered. By generalizing application scenario into the service scenario, possibility of reusing service scenario was found.

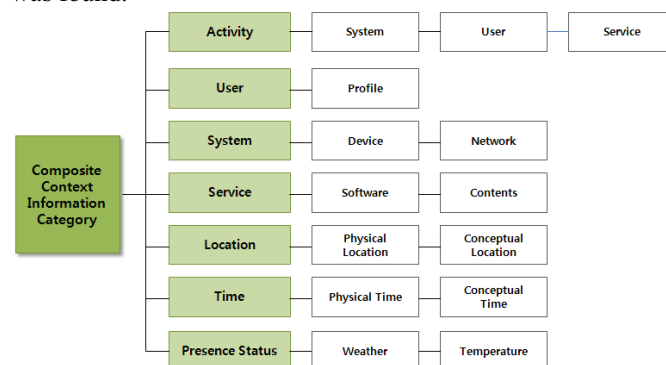


Figure 5. A sample line graph using colors which contrast well both on screen and on a black-and-white hardcopy

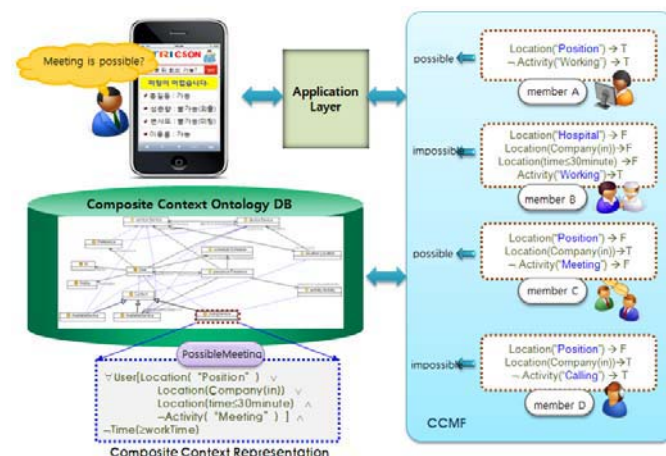


Figure 6. Process of composite context information and provide the service to user according to composite context

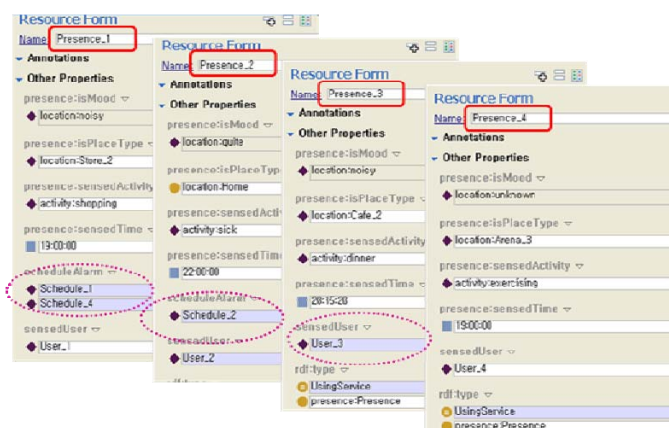


Figure 4 shows presence concept after reasoning by composite context information, we have apply the presence concept which is generate the composite context according each time line. Context aware through ontology and a series

In figure 6, we suggest the process under the composite context information that is receive the each user's situation by composite context, and decide the context to send the user's device, when user have to decide the context that is 'meeting is possible?' finally our suggested system, can support the decision of user.

D. Composite Context Information Process Structure

We have to design the composite context information process structure to implement in systems. Figure 7 can show two feature, one of this, is compare with the existing mash-up-based architecture, and second is structure of composite context information process. Above of all, we can know the benefit in our approach which have a feature the stability and reusability in data frame and application.

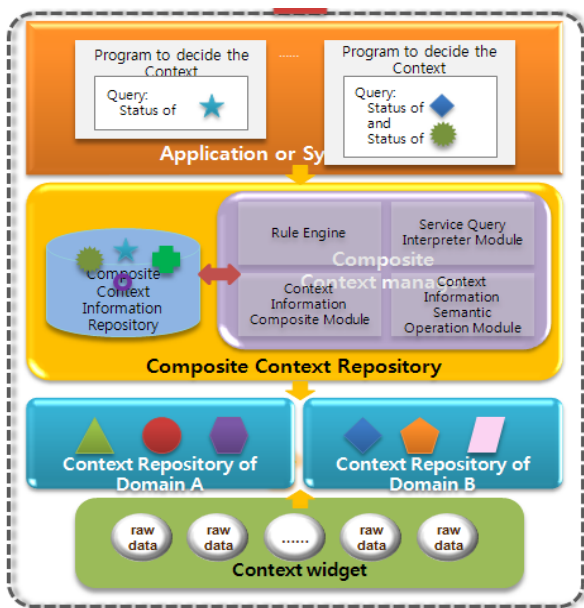


Figure 7. Structure of Composite Context Information Process in Composite Context Repository

We have built two essential components as follows:

- **Composite Context Repository** : it provides a set of terms for describing context knowledge. This ontology allows agents to share a common understanding of the information that they exchange and to reason additional information beyond already known information. This consists of three type of ontology. First, spatial context ontology as a static ontology which is not likely to be changed plays a role in supporting general knowledge of spatial context. Second, sensor data ontology as task ontology is responsible for recognizing features of place where the user may be. This is dynamically generated and discarded in every moment of sensing. Lastly, user profile ontology provides additional information in terms of user preference. It can be quite helpful to infer semantic of the place where the user is. For example, a theater can be either the workplace for a user or just a place for watching movie according to user's profile.

- **Composite Context Information Manager Function** : This is a system that reasons over the semantic model of ontology. To reason about our context ontology, we use a standard ontology language, OWL, and this ontology reasoner provide a set of rules for interpreting the semantic model of OWL and infer necessary information from the knowledge base.

IV. CONCLUSIONS

In order to develop a context-aware application, which can provide more specific information and services about the situation where you may arrange a business meeting as a case study. To develop this application, an approach that selects and combines the strengths of the reviewed methodologies has been adopted. This study, is proposed the composite context information model to provide the knowledge that is determine the path of adaptive service configuration in mobile environment. First of all, our research is defined the composite context information in perspective of this study, and based on this definition, is proposed the approach of master data modeling to configure the composite context information. It is show to can resolved the problem related with data processing, among a lot of difficulties that the proposed context information based system is shown.

The contribution of this paper is twofold. Firstly, we have made comparison and evaluation of existing approaches, and presented the results of the analysis. Subsequently, this analysis results can be used to derive a unified approach from the existing approaches. Secondly, a context framework and ontology has been developed in terms of context management even though there have been many of similar previous approach. The context is the key information that allow service provider to implement reactive services. In future work, we will introduce more sophisticated way of data fusing methodology form many difference feature of data and provide a method for proactive service. It will dramatically facilitate the advantages of using ontology.

ACKNOWLEDGMENT

If needed, one or more (all) author(s) may thank or acknowledge other people's or organization's helps or support.

REFERENCES

- [1] Cortese G., Lunghi M. and Davide F. (2005): Context-Awareness for Physical Service Environments, Ambient Intelligence, IOS Press, pp. 71-96, <http://www.ambientintelligence.org>, 2005.
- [2] Cano J. -C., Manzoni P. and Toh C.-K., "UbiqMuseum: A Bluetooth and Java Based Context-Aware System for Ubiquitous Computing", Wireless Personal Communications, 2006.
- [3] Aziz Z., Anumba C.J., and Law K., "Using Context-Awareness and Web-Services to Enhance Construction Collaboration", Joint International Conference on Computing and Decision Making in Civil and Building Engineering, pp. 3010-3019, June 14-16, 2006.
- [4] Cheverst K., Davies N., Mitchell K. and Friday A., "Experiences of Developing and Deploying a Context-Aware Tourist Guide", The GUIDE Project, ACM MOBICOM 2000.
- [5] Dey A. K., "Providing Architectural Support for Building Context-Aware Applications", PhD Thesis, Georgia Institute of Technology, November 2000.
- [6] Balqis Omar, Tabarak Ballal, "Intelligent Wireless Web Services: Context-Aware Computing in Construction-Logistics Supply Chain",

- [7] [7] Pashtan A., Blattler R., Heusser A. and Scheurmann P., "CATIS: A Context-Aware Tourist Information System", Proceedings of the 4th International Workshop of Mobile Computing, Rostock, June 2003.
- [8] [8] G. Biegel, V. Cahill, V., "A framework for developing mobile, context-aware applications", in Proceedings of the 2nd IEEE Conference on Pervasive Computing and Communication, pp. 361-365, 2004.
- [9] [9] T. Gu, H. K. Pung, D.Q. Zhang, "A middleware for building context-aware mobile services", In Proceedings of IEEE Vehicular Technology Conference (VTC), Milan, Italy, May 2004.
- [10] [10] Harry Chen, Tim Finin, and Anupam Joshi, "An Ontology for Context-Aware Pervasive Computing Environments", The Knowledge Engineering Review, Volume 18, Issue 3, Pages: 197 – 207, 2003.
- [11] [11] Schmidt, A., Aidoo, K.A., Takaluoma, A., Tuomela, U., Van Laerhoven, K., Van de Velde, W., "Advanced Interaction In Context". Lecture Notes in Computer Science, Hand Held and Ubiquitous Computing, pp. 89–101, 1999.
- [12] [12] Karen Henricksen, Jadwiga Indulska, Ted McFadden, Sasitharan, Balasubramaniam, "Middleware for Distributed Context-Aware Systems", R. Meersman and Z. Tari (Eds.): CoopIS/DOA/ODBASE 2005, LNCS 3760, pp. 846–863, 2005.
- [13] [13] Baldauf, M., Dustdar, S. and Rosenberg, F., "A survey on context-aware systems", Int. J. Ad Hoc and Ubiquitous Computing, Vol. 2, No. 4, pp.263–277, 2007.
- [14] [14] Anind K. Dey and Gregory D. Abowd, Daniel Salber, "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications", HUMAN-COMPUTER INTERACTION, Volume 16, pp. 97–166, 2001.
- [15] [15] Gregory Biegel, "A Programming Model for Mobile, Context-Aware Applications", thesis submitted to the University of Dublin, Trinity College, October 2004.
- [16] [16] Davy Preuveneers and Yolande Berbers, "Adaptive Context Management Using a Component-Based Approach", L. Kutvonen and N. Alonstioti (Eds.), DAIS 2005, LNCS 3543, pp. 14–26, 2005.
- [17] [17] Jacob, Ansgar, Müller, Marcus, Kirn, Stefan, "AN EVENT-DRIVEN APPROACH TO DYNAMIC SITUATION DETECTION", 17th European Conference on Information Systems, 2009.
- [18] [18] Anagnostopoulos, C. B., Ntarladimas, Y. and Hadjiefthymiades, S. (2007). "Situational computing: An innovative architecture with imprecise reasoning", J. Syst. Softw. 80 (12), 1993-2014.
- [19] [19] Theodore Patkos, Antonis Bikakis, Grigoris Antoniou, Maria Papadopoulou and Dimitris Plexousakis, "Design and challenges of a semantics-based framework for context-aware services", Int. J. Reasoning-based Intelligent Systems, Vol. 1, Nos. 1/2, 2009.
- [20] [20] Karen Henricksen, Jadwiga Indulska, "Developing context-aware pervasive computing applications: Models and approach", Pervasive and Mobile Computing, vol.2, no.1, pp. 37-64, February 2006.
- [21] [21] Indulska, J. and Sutton, P., "Location management in pervasive systems", CRPITS'03: Proceedings of the Australasian Information Security Workshop, pp.143–151, 2003. S. Chen, B. Mulgrew, and P. M. Grant, "A clustering technique for digital communications channel equalization using radial basis function networks," *IEEE Trans. Neural Networks*, vol. 4, pp. 570–578, July 1993.
- [22] [22] R. J. Vidmar, On the use of atmospheric plasmas as electromagnetic reflectors. *IEEE Trans. Plasma Sci.* [Online]. 21 (3). pp. 876–880. Available: <http://www.halcyon.com/pub/journals/21ps03-vidmar>, Aug. 1992.