Semantic model for context – aware service provision in disadvantaged network environment

Joanna Śliwa, Marek Amanowicz
Military Communication Institute
ul. Warszawska 22 A
05-130 Zegrze, Poland

{j.sliwa, m.amanowicz}@wil.waw.pl

ABSTRACT

The use of ontologies can facilitate many processes in C4I systems. They can be used to provide unambiguous description of data sent among systems as well as automate realization of services in dynamic NEC environment. The use of ontologies has been also proposed in the area of mobile computing, where the ability to react to dynamic changes of the environment with minimal human intervention is a fundamental requirement. A common element in the architecture of ubiquitous applications is a proxy, an element in charge of executing a number of content adaptations on behalf of one or several client applications running on mobile devices. These adaptations are triggered by specific conditions involving the mobile devices on which the applications execute. This article describes proposal of the semantic model for context – aware service provision in disadvantaged environment that is used to dynamically select adaptation actions performed on SOAP messages flowing from the service to the client entity.

1.0 INTRODUCTION

Modern coalition operations are conducted in a dynamic environment, usually with unanticipated partners and irregular adversaries. In order to act successfully they need technical support that gives modularity and flexibility in connecting heterogeneous systems of cooperating allies. To support such co-operation in the NATO community, the Service Oriented Architectures (SOAs) [1] are recommended as the crucial Network Enabled Capability (NEC) enabler [2-4].

The most mature implementation of SOA, recommended by NATO and widely applied in the commercial sector, are Web Services (WS) [5]. WSs are described by a wide range of standards that deal with different aspects of their realization, transport, orchestration, semantics, etc. They provide the means to build a very flexible environment that is able to dynamically link different system components to each other. These standards are based on the eXtensible Markup Language (XML) and have been designed to operate in high bandwidth links. XML gained wide acceptance and became very popular for the reason that it solves many interoperability problems, is human- and machine-readable and facilitates the development of frameworks for software integration, independent of the programming language. Nevertheless it undoubtedly adds significant overhead, both in terms of necessary computation power and consumption of network resources while being transported. The challenge is therefore to apply SOA in low bandwidth tactical communications systems, which usually cope with high error rates and frequent disruptions. Such networks are usually referred to as disadvantaged ones.

Theoretically, the Information and Integration Services (IIS) infrastructure that constitutes the basis for creating information sharing capabilities and building the Common Operational Picture is independent of the underlying communication infrastructure. In fact, comparing stationary communication networks and radio networks, there are many differences in functionality, assurance, usability and utility of different IIS components. In order to take the full advantage of SOA and web services on tactical level there are necessary edge functionalities (e.g. Edge proxy) that will be able to adapt services traffic to the possibilities and current performance of the tactical networks [6]. There must be provided a method of adapting the WS realization, to enable information distribution to users located in tactical networks, provided that the conditions posed by the client and the service producer are met. Such a situation is visible e.g. when a military unit works in a low bandwidth wireless network and cannot get a complete Common Operational Picture of its forces. This service should be adapted by the content and/or transmission means and made available even with a smaller quantity of information, but with the data crucial for the unit's operation in the field.

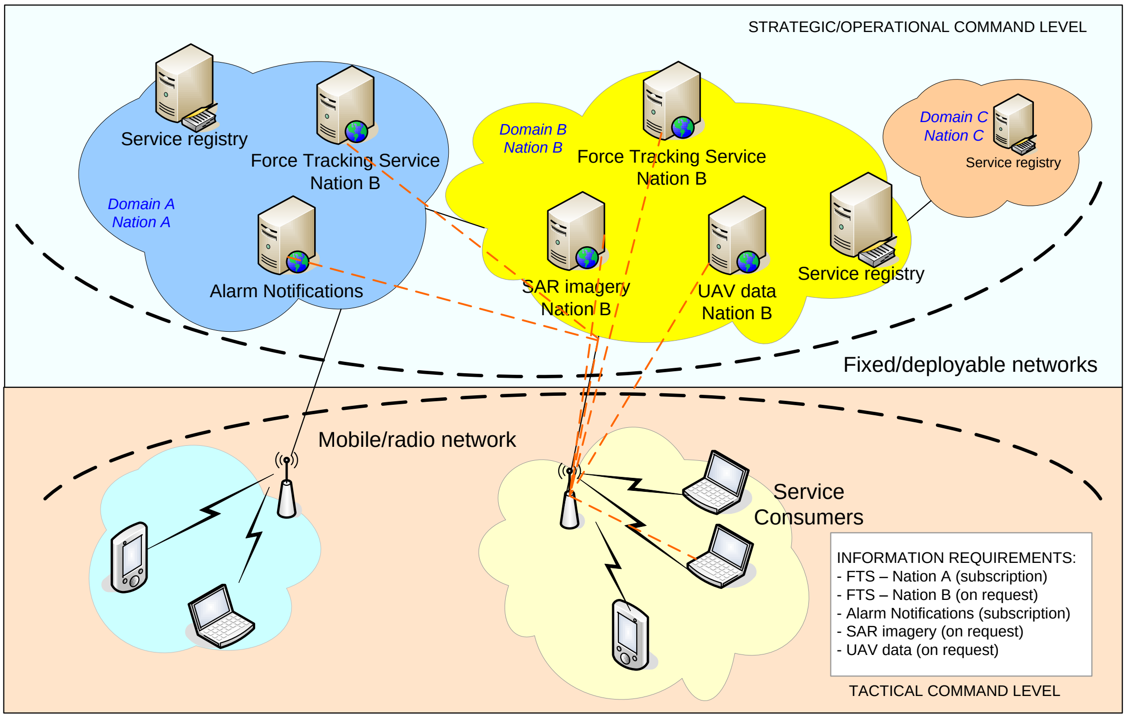

Figure 1: Client – server relations in military network

The services at the first stage of the Networking and Information Infrastructure (NII) development are mainly located on high echelons of command – strategic and operational. They are used for planning purposes and provide good basis for creating situational awareness and self–synchronization of cooperating forces. However information in the system is being exchanged vertically and horizontally among mission participants in order to fulfil their tasks, act faster and make reliable decisions (see Figure 1). Users at the lowest command levels need in particular information about the location and status of their and allied forces as well as about the enemy ones. This information from the Force Tracking Systems is available at the operational level but not always accessible for the lower level commanders. They are usually located in tactical communication systems that are known from radio communications with scarce network resources in terms of high delay, error rate, and limited bandwidth. What is more, they are equipped with mobile terminals that have limited computational and software resources as well as limited battery power. That makes it difficult to provide the user with the same service functionality as provided to the users at operational and strategic command levels. The tactical user is very often not able to receive nor process a big amount of data. The solution that this article focuses on is therefore to enable the client to use the service in a limited way (with limited number of information provided or provided by a different mechanism) and adapt the service provision mechanism to the client's software and hardware possibilities.

2.0 CONTEXT – AWARE SERVICE PROVISION

Context – aware applications refer to a general class of mobile systems that can sense their physical environment, and adapt their behaviour accordingly. They derive from the ubiquitous (or pervasive) computing concept was presented in 1991 by Mark Weiser [7] who set its foundation. This concept developed for the commercial applications began the new field of interest of many researchers where the area of context-aware applications became an important part.

In context – aware service provision it is generally important where the client is, what are his actions/duties, what terminal is he using, what resources are nearby, etc. [8]. In many applications the most important aspect is location but this can be extended to include different characteristics (user actions, device, surrounding environment, etc.). Context recognition allows users to take full advantage of the local capabilities within a given environment, and be able to seamlessly roam between several environments, choose different services, even as the defined context change.

The idea of context – aware service provision was used in the development of the Adaptation Framework for Web Service provision in disadvantaged environment (AFRO). It is aimed at improving successability of SOAP web services invocation in tactical environment, which is characterized by dynamic changes in throughput, error rate and delay. Successful service invocation in this case is understood as the possibility to deliver response message requested by the client from the target service.

AFRO follows the assumption that in order for a web service to function more efficiently it is necessary to minimize the amount of data transmitted to the user. The actual traffic flow related to web services' interactions is burdened with the XML overhead which greatly limits communication link goodput. It is highly recommended therefore:

- to improve encoding efficiency, i.e. enhancing the ratio of the user data to the management data in the SOAP message, and
- to reduce the number of unnecessary data (or data that cannot be consumed) to the users of degraded networks.

Limiting the size of traffic flow to the users of wireless networks will improve the successability of web service calls and will support users with information crucial for their operation in the battlefield.

Message adaptation actions can be therefore twofold:

- lossless – e.g. actions that improve message encoding enabling the consumer's side to decode it without losing any of the data or without the transport protocol change (e.g. HTTP to MMHS), and
- lossy – cutting out information that the user agrees to be filtered out.

Selection of appropriate adaptation action is not a trivial task and needs to be based on several types of information. First of all, adaptation does not have to be performed when the connections are stable, network has high bandwidth, acceptable delay, no losses, and therefore, does not provide limitations for web service provision. The information about the network state is important in order not to spoil time on unnecessary actions.

The second important aspect is necessity to take into account users' preferences in terms of adaptation. They will be included in, so called, user profile, within which the user will state his adaptation preferences, and device profile, which defines his terminal's software and hardware possibilities. This set of information is necessary for selecting the actions that would meet the user intentions and, at the same time, would not make it impossible for the user terminal to receive and decode the message. This results from the fact that when the message is received at the user terminal it is firstly processed by the software libraries installed on it. Existence of particular software libraries implies therefore possibility of particular message encoding actions. Additionally, terminal information can help in parameterizing images and video streams that would be directed to the user working on a particular device.

Such an approach makes it necessary to provide a mechanism for provisioning and then efficiently using information about the user, his terminal, the network and service invoked. This problem has been defined as the need to identify the context of the service call. It has been proposed in the form of ontology that allows to clearly define parameters of entities taking part in the information distribution process and then, on the basis of the set of rules and the rule engine, efficiently support the decision process enabling to take adaptation actions improving service successability.

On the basis of previous considerations the architecture of the Adaptation Framework for Web Service Provision (AFRO) has been proposed (see Figure 2). It bases on the Decision Support Engine that uses information about the context of the service call as the input data, and, on the basis of ontology rules, defines the adaptation actions to be triggered on the SOAP body and SOAP attachments by the Adaptation engine. Such a modified SOAP message has smaller size than the original one and as such, is sent to the requester.

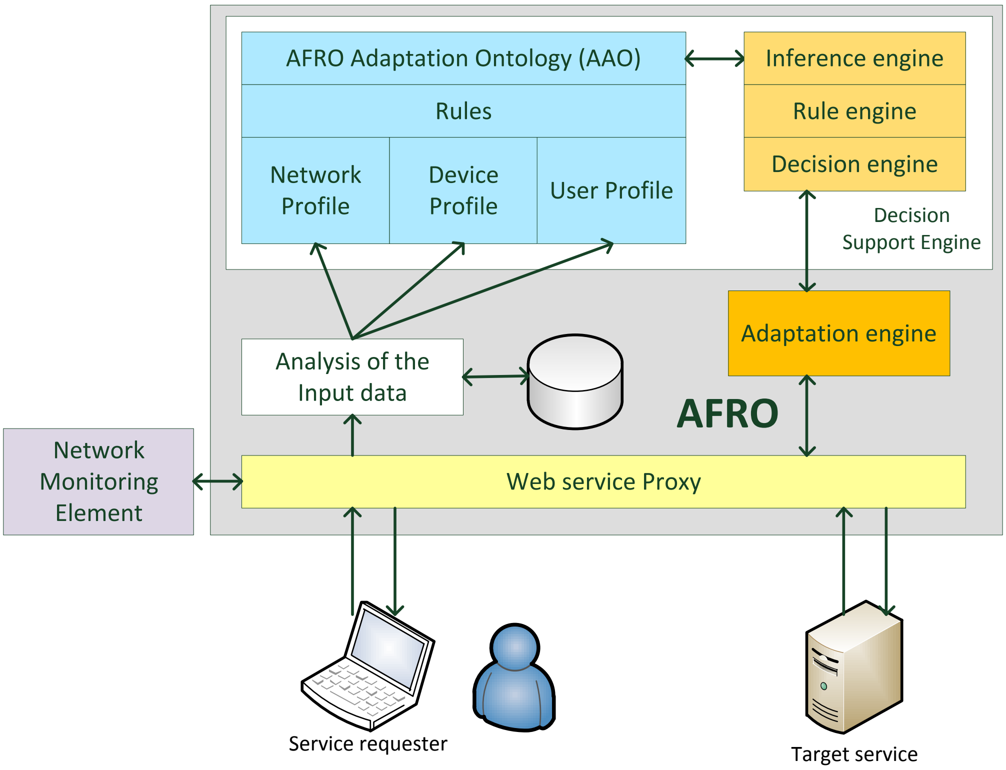

Figure 2: AFRO architecture framework

The ontology proposed and the rule engine strongly support dynamic selection of adaptation actions appropriate for the user. They are used by the Decision Support Engine that returns in response a set of actions. These actions derive from the Proxy functionality. They can be embedded (e.g. take the form of the Adaptation engine, see Figure 2) or, taking different approach, distributed. The latter one can be implemented using SOA services orchestration. After the Proxy selects appropriate actions for the user, it searches for the services that will provide appropriate mediation (will carry out the action).

Whatever approach to Proxy implementation one can take, the application of the Decision Support Algorithm and the proposed adaptation ontology (AAO) will supply him with the dynamic selection of actions to be taken.

It is also assumed that the Proxy will make use of information provided by external element – Network Monitoring Element that will support it with information about currently observed network performance on the link to the user. This performance information (in terms of throughput, delay and error rate) will be used by the decision support algorithm. In case the network is categorized as disadvantaged, the Proxy will make the modifications stronger, decreasing the amount of information that is sent to the user (in terms of image modifications), however making it more probable to be transferred to the consumer.

2.1 Reflecting user requirements in the service call context

One of the elements of the proposed Method is context of the service call. In case of the AFRO proxy it is perceived as a collection of information:

- about the user:
  - What modifications of the SOAP messages' content is the user willing to accept?
  - What device is he using as his end terminal?
  - What access network is he using?
- the device:
  - What are the characteristics of the device hardware (resolution of the screen, CPU frequency)?
  - What are the characteristics of the device software (operating system, supported libraries)?
- the service:
  - What is the service description?
- and underlining network:
  - What is the network type?
  - What is the current link performance?

The reason for the dynamic adaptation to be based on the pre-distributed information is that the user – from the point of view of his activities – may not wish the mechanism to modify contents of the message and modify the attachment (resize, compress, decrease colour depth). In order for the non-standard XML encoding to be used at the receiver, the device must be equipped with appropriate libraries. It is very often an issue in mobile devices that use limited operating systems and limited set of libraries and do not support software implementations regularly used in laptops or PCs. The environment the adaptation framework is to be used in assumes utilization of mobile hand-held devices the configuration of which (software and hardware) is important in terms of successful web services adaptation.

The context of the service call has been modelled semantically with the Web Ontology Language, (OWL DL [9]) which is the most powerful ontology description language and promising in terms of further processing, rule enforcement and inference [10].

The context information has static and dynamic elements. It generally consists of: user context (adaptation preferences – static), device profile (static), service context (QoS profile – static), network context (Link performance – dynamic).

2.2 Application of rules

For the purpose of selecting the adaptation actions the decision engine uses the AFRO Adaptation Ontology (AAO) describing all the actions that can be taken by the proxy, reflecting the user preferences. In order to make use of the adaptation ontology a set of rules has been defined. Rules are important in OWL to state facts about instances of classes.

The rules in AFRO define requirements for particular actions based on information that are provided in the context of the service call. They have been defined using the Semantic Web Rule Language (SWRL) [11], combining sublanguages of the OWL DL and Lite with those of the Rule Markup Language. OWL Full constructs, such as classes, property values, are not supported by this language so that it does not support direct reasoning about classes or properties. It is not possible to write a rule that, for example, deduces some new knowledge based on the fact that one class is a direct subclass of another. For the same reason, RDF (Resource Description Framework) [12] or RDFS (RDF Schema), or OWL constructs such as owl:Class or owl:DatatypeProperty, cannot be used in rules.

This is because the OWL is based on the Open World Assumption (OWA) that states that anything might be true unless it can be proven false. Open World Assumption states therefore that everything we don't know is undefined. This is contradictory to the Closed World Assumption that refers to "everything we don't know is false". That is why according to OWA we cannot specify that a fact f(x,y) can be true when x and y are instances and there are other properties or facts defined. In order to provide additional facts to ontology, that will add e.g. available actions to be performed for particular call, the rules are proposed.

2.3 Gathering context data

AFRO proxy can work in the request – response mode which resembles the situation when the user invokes the target service through the AFRO proxy and publish – subscribe mode – when the user subscribes to data flowing from the target service. In both cases before the actual service data will be distributed, the static context information should be pre-distributed to the AFRO proxy.

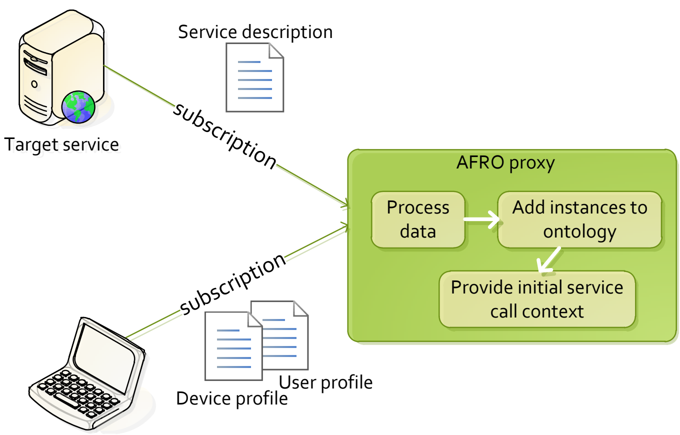

Figure 3: Pre-processing of the data gathered during the subscription process

The dynamic elements of the context should be gathered at run time by agent entities and forwarded to the proxy. In this case they relate to the current link performance.

The architecture of the AFRO Proxy assumes support for dynamically selecting web service adaptation actions in run time. It assumes the SOAP body and SOAP attachment adaptations that limit the size of SOAP messages. The actions taken by the Proxy can be internally implemented, or can be served according to SOA concept, by external entities. AFRO Proxy architecture enables to add additional plugins that can be used for further actions. What is more, since the "heart" of the method is the decision support engine that bases on ontology, a set of rules and the rule engine, it can be easily used in the process of services composition, that would make use of particular services as adaptation actions, that would return adapted messages in return.

3.0 AFRO ADAPTATION ONTOLOGY (AAO)

The main subject of this article is description of the semantic model of the context of the service call. It should reflect all the elements of the web service call, which can be used in the decision-making process to select appropriate adaptations. It must characterize static elements that can be defined before service call is executed, as well as dynamic elements that depend on the type of request, type of response and temporary network performance parameters.

Among many known methods of context modelling, semantic description using ontologies gives the most satisfactory results [10]. In general, ontology describes formally a domain of discourse. In the computer engineering, ontologies are usually employed to provide semantic interoperability among cooperating systems and increase the level of automatic reaction to events (e.g. sent/received information). In this Article, ontology is used to define the context of the service call, making it possible to further process it and make decisions about the adaptation actions.

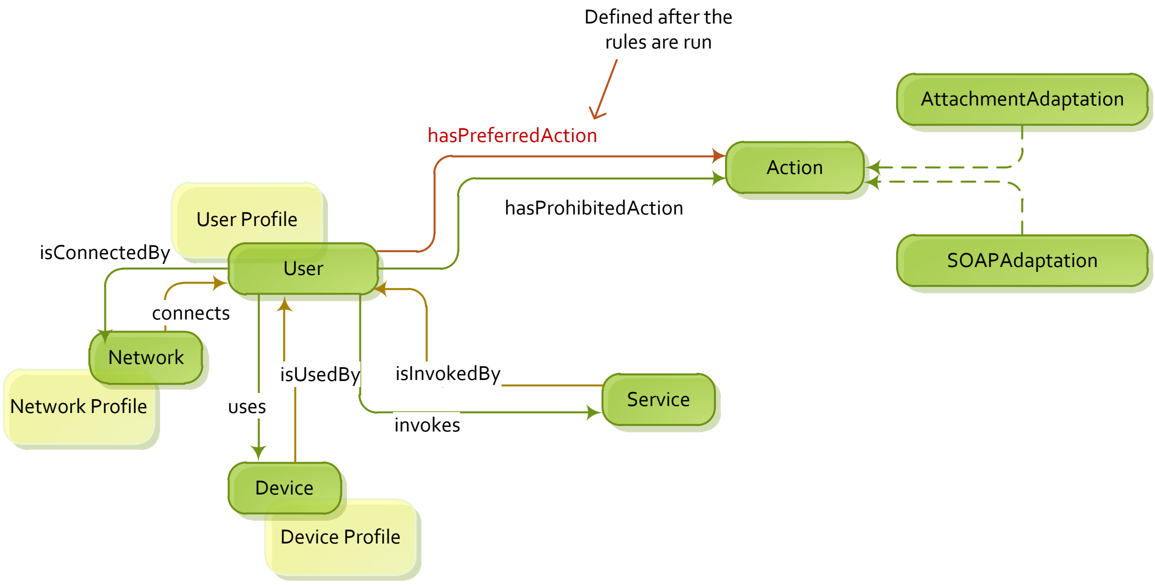

Figure 4: AFRO Adaptation Ontology

For the purpose of the AFRO proxy the context has been defined using the most well-known and semantically rich languages, i.e. RDF (Resource Description Framework), RDFS (RDF Schema) and OWL (Web Ontology Language). The context is created based on User profile, Device profile, Network profile and Service profile. AAO was developed according to the IOEM ontology engineering ontology [14].

These four entities are connected by the AAO upper ontology (see Figure 4). According to this ontology the `User` is connected to the infrastructure by particular `Network` and uses as its end terminal particular `Device`. It invokes the `Service`. These four entities form the context of the service call. Every User has a set of prohibited and preferred `Actions`. They can be set at the initial stage of gathering user data (the user may indicate them directly) or automatically selected by the ontology rules on the basis of information about the `Device`.

This upper ontology presents relationships among the main entities of the model. Behind the `User`, `Network` and `Device` there are defined profiles that describe their main characteristics.

The information about the User and his device, received by the AFRO Proxy, after being analysed, is saved in the AFRO Adaptation Ontology (AAO) instances. The User and Device profiles are therefore used to unambiguously express user preferences in terms of adaptation actions and express his device limitations. AAO enables to express the Display limitations, CPU frequency limitations, list unsupported MIME types [13] (e.g. particular formats of images or video) and encoding mechanisms.

After the user logs in, his every request is perceived as a Service call. On the basis of rules defined in the Proxy, appropriate adaptation actions are selected. An exemplary rule defining the ChangeResolutionAction as preferred for the user when his device has low CPU is as follows:

uses(?x, ?y)^hasHWlimitations(?y, ?z)^LowCPU(?z) -> hasPreferredAction(?x, ChangeResolutionAction)

The Action class is divided into two subclasses: SOAPAdaptationAction and AttachmentAdaptationAction. They allow for creating different actions that the Proxy can provide (or can invoke in an external entity).

3.1 User profile

The user profile stores all the information that is gathered by the AFRO proxy during the user subscription. It describes all the user requirements in terms of actions:

- that he is willing to enforce (hasPreferredActions object property that indicates actions preferred), e.g. decrease colour depth modification of the message attachment is necessary because the terminal has grayscale display;
- that he is willing to prohibit (hasProhibitedActions object property that indicates actions prohibited), e.g. the user does not allow to make any modifications of the attachment, or the device does not support particular action,
- and gives details related to the messages' content adaptation (hasAdaptationPreferences object property).

This makes the user able to influence the AFRO adaptation actions e.g. according to his role and duties. The same object properties are also used by the decision support algorithm to dynamically indicate appropriate adaptation actions. The user profile also indicates the device the user is currently exploiting (Device), the network it is connected by (Network) and gives basic information for the user's identification. The user profile information is related to the User class in the AAO ontology (see Figure 5).

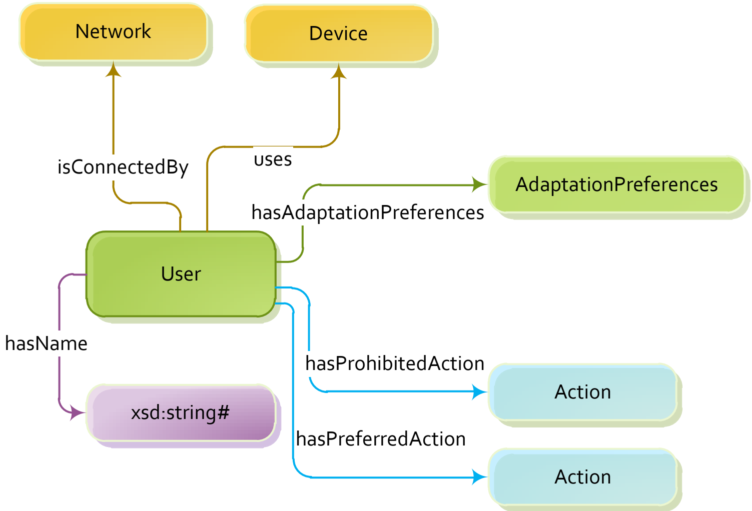

Figure 5: User ontology – first level of class taxonomy

The adaptation preferences (see Figure 6) that the user can set up define parameters regarding attachment adaptation actions. In case of the actions taken into account in the Article there have been proposed the following preferences used for SOAP with attachments (images): preferred resolution of the image, preferred quality of the image, preferred colour depth of the image. The values of these preferences will be taken into account when performing the adaptation. When the invocation is performed in the constrained network, these preferences will be taken directly. When the client that performs invocation is located in the degraded network, the values of adaptation preferences will be calculated for him separately.

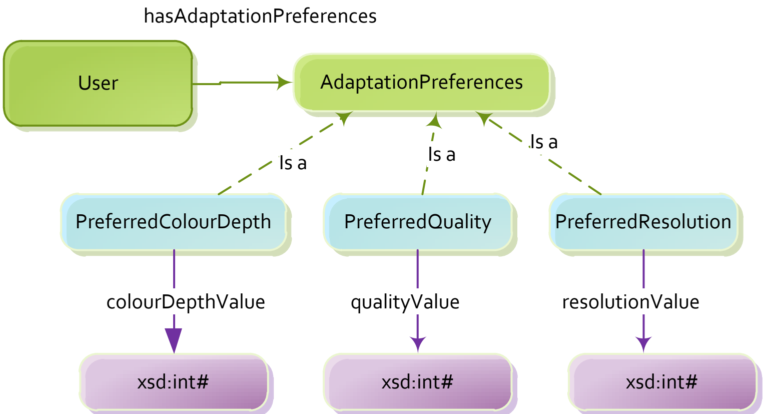

Figure 6: User Adaptation Preferences according to the AFRO user ontology

3.2 Device profile

Device profile is based on the Composite Capabilities/Preferences Profile (CC/PP) [15], which is the W3C recommendation and provides description of device capabilities and user preferences. It is used by the mobile devices' manufacturers that provide descriptions of their products in CC/PP vocabulary. Selecting this description model enables keeping interoperability with existing solutions and gives the opportunity of the opposite. The proposed in AAO Device profile based on CC/PP and defines additional elements (see Figure 7). It defines Device Display Properties, Encoding Properties, Hardware Properties and MIME Types Supported.

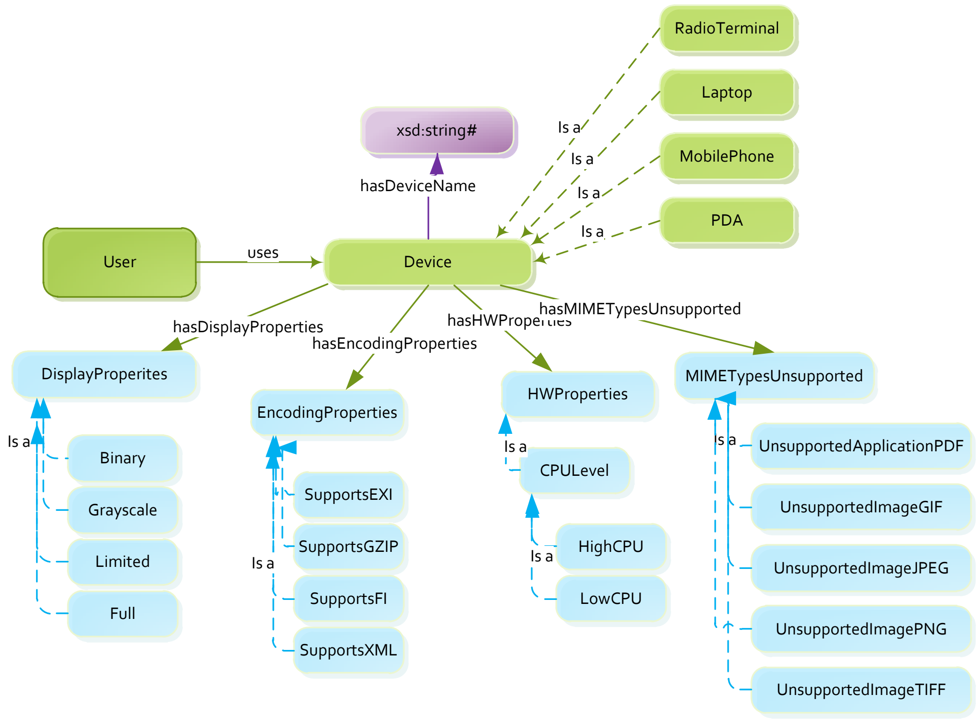

Figure 7: Device profile according to AAO

3.3 Network ontology

Network ontology (see Figure 8) describes the performance of the network connection between the user and the AFRO Proxy. It is defined by the `NetworkType`, which can be `VeryGoodNetwork`, `ConstrainedNetwork` and `DegradedNetwork`. Network profile is dynamically created on the basis of information received from the network monitoring element when the user requests data (sends request message). Additionally, each network can be classified as `CableNetwork` or `WirelessNetwork`.

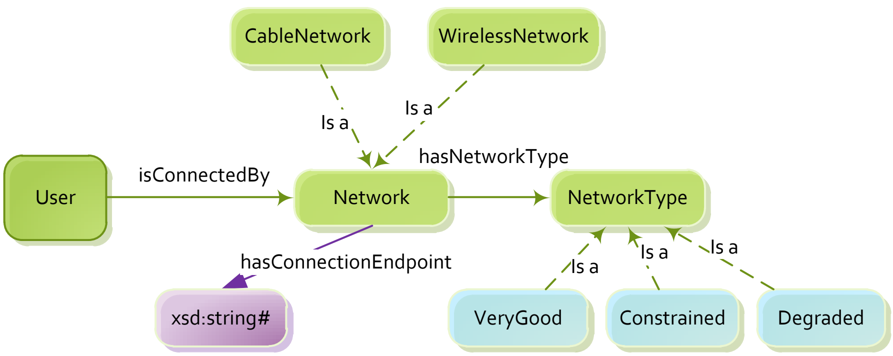

Figure 8: Network ontology

Network state is determined on the basis of information received from external entity – Network Monitoring Element. It is assumed that this element will monitor the link to the user at supply the AFRO Proxy with information about the current link performance in terms of currently observed Throughput, Packet Error Rate (PER) and Delay of packets transmission. This information will be checked whenever the decision process is to be performed.

Preliminary researches that have been carried out in [16] proved that web services efficiency is decreased in particular ranges of values of these three QoS parameters. On the basis of obtained results there have been defined three levels of the network state, i.e. “very good”, “constrained” and “degraded” network state defined in the following way:

- Degraded network state level is when PER ≥ 10% or delay ≥ 300 ms or throughput <10 kbps.
- Very good network state is when PER < 5% and delay < 100 ms and throughput ≥ 2000 kbps.
- The remaining values of link performance parameters define the network as constrained.

3.4 Adaptation actions

Adaptation ontology has been designed to describe all the mechanisms that the AFRO proxy can use to adapt messages to be delivered to the user. The Action class has two direct subclasses: AttachmentAdaptation and SOAPAdaptation. They represent all the mechanisms available for the AFRO proxy, which are:

- Class: GZIPCompression,

  superclass: SOAPAdaptation,
  individuals: GZIPcompressAction,
  comment: Compresses the message with the GZIP algorithm,

- Class: EXIEncoding,

  superclass: SOAPAdaptation,
  individuals: EXIencodeAction,
  comment: Encodes the message with EXI,

- Class: FIencoding,

  superclass: SOAPAdaptation,
  individuals: FIencodeAction,
  comment: Encodes the message with FI,

- Class: Filtering,

  superclass: SOAPAdaptation,
  individuals: XPathFilteringAction,
  comment: Enables to filter out the SOAP message,

- Class: ChangeResolution,

  superclass: AttachementAdaptation,
  individuals: ChangeResolutionAction,
  comment: Changes resolution of the images sent in the response,

- Class: DecreaseColourDepth,

  superclass: AttachementAdaptation,
  individuals: DecreaseColourDepthAction,
  comment: Decreases colour depth of the images,

- Class: DecreaseQuality,

superclass: AttachementAdaptation,
individuals: DecreaseQualityAction,
comment: Changes quality of the JPEG images increasing their compression ratio,

- Class: DiscardAttachment,

  superclass: AttachementAdaptation,
  individuals: DiscardAttachmentAction,
  comment: Discards attachment from the message.

The adaptation actions can be LosselessAdaptationActions and LossyAdaptationActions. The lossless are e.g. compression or binary coding of SOAP messages. Lossy would be e.g. filtering of SOAP messages and all Attachment Adaptation Actions.

The AttachmentAdaptationActions are the actions that can manipulate the SOAP attachments. In this Article there has been considered example of JPEG image modifications, but other actions can be defined for further purpose of using the AAO. In the image adaptation actions one can find therefore decrease colour depth action, compression action and decrease colour depth action. These are all lossy adaptations that result in decreasing the level of information that is then transferred to the client.

The adaptation ontology can be further expanded as additional components reflecting actions available in the AFRO proxy will be introduced.

3.5 Rules supporting the decision process

The TBox statements define properties about entities, however they cannot define conditional statements, e.g. If a Student studies Maths then he is a Maths Student. For this purpose it is recommended to use rules and rule engine that allow for adding certain facts to the knowledge base on the basis of existing axioms.

Rules are of the form of an implication between an antecedent (body) and consequent (head). Their meaning can be read as: whenever the conditions specified in the antecedent hold, then the conditions specified in the consequent must also hold. In relatively informal "human readable" format:

```
antecedent (body) ⇒ consequent (head).
```

Both the antecedent (body) and consequent (head) may consist of zero or more atoms. An empty antecedent is treated as trivially true (i.e. satisfied by every interpretation), so the consequent must also be satisfied by every interpretation. When a consequent is empty, it is treated as trivially false (i.e. not satisfied by any interpretation), so the antecedent must also not be satisfied by any interpretation.

When both antecedent and consequent are conjunctions of 1 – n atoms the rule takes the following form: a1 ^ ... ^ an. Variables are indicated using the standard convention of prefixing them with a question mark (e.g., ?x). Using this syntax, there can be defines a rule asserting that if a parent (x2) has a child (x1) and a brother (x3), the brother is an uncle to the child, i.e.:

```
hasParent(?x1,?x2) ∧ hasBrother(?x2,?x3) ⇒ hasUncle(?x1,?x3)
```

The rules can be defined using a few formal languages, e.g. Jess rule language, JessML, RuleML (Rule Markup Language), SWRL (Semantic Web Rule Language). Due to the easiness of defining and processing rules in SWRL, this language has been selected to be used in the Article. It uses the human-readable syntax as presented above together with the abstract and XML syntax.

The AAO ontology has been enriched with the set of rules that enable to define actions that can be performed by the Proxy. The use of rules automates the decision process. Moreover, since the rules are not hard-coded, the semantic programming tools enable to easily define additional rules or modify existing ones.

Practical tests of the adaptation actions proposed for the AFRO proof of concept [16] was the basis for defining decision rules for adaptation decision support algorithm that would match user preferences and suit best for available network resources. On the basis of the context of the service call and the set of rules additional axioms of available adaptation actions are added to the Knowledge Base. For instance, if the user's device has limited display properties, the Proxy should decrease Colour depth of the image attachment:

```
uses(?x, ?y)^hasDisplayProperties(?y, ?z)^Limited(?z)->
hasPreferredAction(?x, DecreaseColourDepthAction)
```

Taking into account user device limitations there has been defined the following adaptation rules:

1. Rule 1 – Discard PDF
   Description: If the user device does not support particular format of the MIME attachment (e.g. PDF), the attachment should be discarded.
   Rule content:
   ```
   uses(?x, ?y)^hasMIMETypeUnsupported(?y, ?z)^
   UnSupportedApplicationPDF (?z) -> hasPreferredAction(?x,
   DiscardPDF)
   ```
2. Rule 2 – Discard GIF
   Description: If the user device does not support particular format of the MIME attachment (e.g. GIF image), the attachment should be discarded.
   Rule content:
   ```
   uses(?x, ?y)^hasMIMETypeUnsupported(?y, ?z)^ UnSupportedImageGIF
   (?z) -> hasPreferredAction(?x, DiscardGIF)
   ```
3. Rule 3 – Discard JPEG
   Description: If the user device does not support particular format of the MIME attachment (e.g. JPEG image), the attachment should be discarded.
   Rule content:
   ```
   uses(?x, ?y)^hasMIMETypeUnsupported(?y, ?z)^ UnSupportedImageJPEG
   (?z) -> hasPreferredAction(?x, DiscardJPEG)
   ```
4. Rule 4 – Discard PNG
   Description: If the user device does not support particular format of the MIME attachment (e.g. PNG image), the attachment should be discarded.
   Rule content:
   ```
   uses(?x, ?y)^hasMIMETypeUnsupported(?y, ?z)^ UnSupportedImagePNG
   (?z) -> hasPreferredAction(?x, DiscardPNG)
   ```
5. Rule 5 – Discard TIFF
   Description: If the user device does not support particular format of the MIME attachment (e.g. TIFF image), the attachment should be discarded.
   Rule content:
   ```
   uses(?x, ?y)^hasMIMETypeUnsupported(?y, ?z)^ UnSupportedImageTIFF
   (?z) -> hasPreferredAction(?x, DiscardTIFF)
   ```
6. Rule 6 – lowCPU
   Description: If the user device has CPU with frequency lower than 1000 MHz it is a low performance CPU, utilization of which will result in high image processing times. If the image attachment has higher resolution than the device display, it should be resized.
   Rule content:
   ```
   uses(?x, ?y)^hasHWlimitations(?y, ?z)^LowCPU(?z) ->
   hasPreferredAction(?x, ChangeResolutionAction)
   ```

7. Rule 7 – Binary
   Description: If the user device has display unit that enables to visualize only binary colours (e.g. black-white), the attached image will be displayed only in binary colour depth. The image attachment, before being sent should therefore have the colour depth set to binary.
   Rule content:
   ```
   uses(?x, ?y)^hasDisplayLimitations(?y, ?z)^Binary(?z)
   -> hasPreferredAction(?x, DecreaseColourDepthAction)
   ```
8. Rule 8 – GreyColour
   Description: If the user device has display unit that enables to visualize only grey scale, the attached image will be displayed only in grey scale. The image attachment, before being sent should therefore have the colour depth set to grey scale.
   Rule content:
   ```
   uses(?x, ?y)^hasDisplayLimitations(?y, ?z)^Grayscale(?z)->
   hasPreferredAction(?x, DecreaseColourDepthAction)
   ```
9. Rule 9 – LimitedColour
   Description: If the user device has display unit that enables to visualize only limited amount of colours, the image attachment, before being sent should therefore have the colour depth decreased appropriately.
   Rule content:
   ```
   uses(?x, ?y)^hasDisplayLimitations(?y, ?z)^Limited(?z)->
   hasPreferredAction(?x, DecreaseColourDepthAction)
   ```
10. Rule 10 – DecreaseQuality
    Description: The images, when compressed appropriately, do not decrease their readability significantly. When send to mobile devices' users they should be compressed.
    Rule content:
    ```
    User(?x)->
    hasPreferredAction(?x, DecreaseQualityAction)
    ```
11. Rule 11 – GZIP
    Description: If the user terminal has the GZIP compression support, the GZIP compression is possible to be performed.
    Rule content:
    ```
    uses(?x, ?y)^supportsEncoding(?y, ?z)^SupportsGZIP(?z)->
    hasPreferredAction(?x, GZIPcompressAction)
    ```
12. Rule 12 – EXI
    Description: If the user terminal has the EXI encoding support, the EXI encoding is possible to be performed.
    Rule content:
    ```
    uses(?x, ?y)^supportsEncoding(?y, ?z)^SupportsEXI(?z)
    -> hasPreferredAction(?x, EXIencodeAction)
    ```
13. Rule 13 – FI
    Description: If the user terminal has the FI encoding support, the FI encoding is possible to be performed.
    Rule content:
    ```
    uses(?x, ?y)^supportsEncoding(?y, ?z)^SupportsFI(?z)->
    hasPreferredAction(?x, FIencodeAction)
    ```

4.0 ADAPTATION DECISION SUPPORT ALGORITHM

The Adaptation Decision Support Algorithm is run by the Decision Support Engine and consists of two phases:

- Initial phase – creating InitialServiceCallContext – on the basis of the data provided by the user (static data) (see Figure 9),
- Final phase – creating FinalServiceCallContext – on the basis of dynamic data from the target service (response message) and from network monitoring element.

The first phase will analyse the user and device profiles and generate set of actions that are prohibited by the user or not supported by his terminal as well as recommended actions that should be considered on the basis of user terminal limitations (deriving for instance from the terminal limitations).

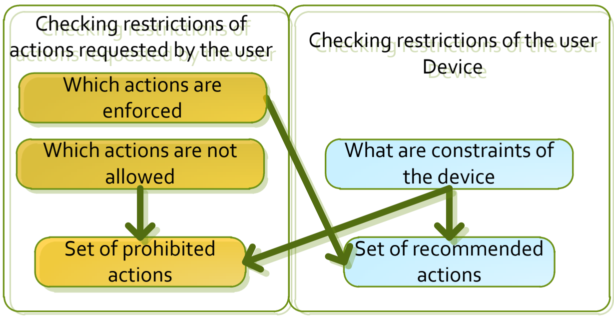

Figure 9: Initial phase of the decision support Algorithm

AFRO Proxy makes decisions about adaptation actions to be performed on the message received from the target service using the Adaptation Decision Support Algorithm. It is based on the Afro adaptation ontology (AAO) and rules defined for the purpose of AFRO (see Figure 10).

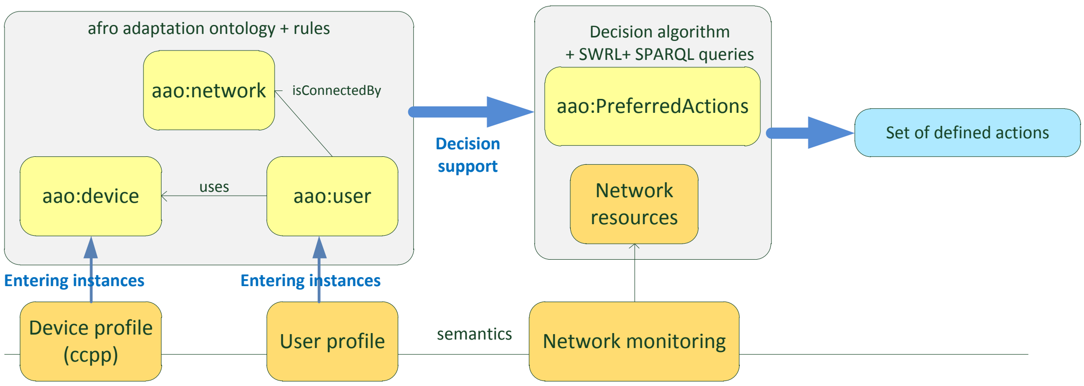

Figure 10: High level view on the AFRO Adaptation decision algorithm

5.0 VALIDATION

Ontology evaluation is a process aimed at validation and verification of an ontology in terms of its scope, consistency and expressiveness [14].

The scope of the AFRO adaptation ontology (AAO) has been set up by the problem it is designed to solve. It is aimed at supporting the dynamic selection of adaptation actions taken on the SOAP messages exchanged between the web service client and server. It defines:

- entities that take part in the service invocation as classes (User, Device, Network, Service, Action class),

- relationships among entities as object properties (connects ↔ isConnectedBy, hasAdaptationPreferences, hasDeviceProperties, hasPreferredAction, hasProhibitedAction, usedBy ↔ uses, hasNetworkType, isInvokedBy ↔ invokes),
- characteristics of entities as data type properties (userName, deviceName, qualityValue, resolutionValue, colourDepthValue).

The TBox ontology model describes relationships among defined entities. On its basis knowledge about the service call context (defined in ABox entries) is collected. After each user registers to the proxy, the knowledge about the user preferred, prohibited actions and his device properties are saved in ABox entries. This allows to set the Initial Service Call Context (ISCC). After the network state is checked, the final AFRO defined actions set (ADA) is defined.

The AAO is the basis for running the decision support algorithm and setting the actions that should be performed by the AFRO Proxy.

Ontology rules defined for the purpose of selecting the actions take into account the following cases:

- the terminal does not support particular file format → the attachment is discarded (rule 1 – 5),
- the terminal supports particular encoding techniques → the encoding actions is added to the list of preferred (rule 11 – 13),
- the terminal has too low CPU frequency (processing power) → big images will be difficult to be processed – change image resolution (rule 6),
- the terminal has limited colour display – decrease colour depth (rule 7 – 9),
- the terminal is connected by disadvantaged network link (general rule – true for all cases) – decrease quality (rule 10).

Moreover, the preferred and prohibited actions that the user defined are also taken into account. They may derive from the role of the user and his duties at the battlefield.

The AAO defines all entities that are necessary to take appropriate adaptation decision and enables to automatically select appropriate adaptation actions. Its scope covers the required level of detail in describing the entities and relationships among them taken in the initial phase of ontology development. It covers so called competency questions [14] defined for the purpose of AAO. Additionally, the set of rules monitor all basic terminal characteristics that may influence usability of messages delivered to the user.

The second ontology evaluation step consists in checking the ontology consistency. According to [17] ontology is consistent (also called satisfiable) when it does not contain a contradiction. The lack of contradiction can be defined in either semantic or syntactic terms. The syntactic definition states that a theory is consistent if there is no formula P such that both P and its negation are provable from the axioms of the theory under its associated deductive system.

The ontology model that contains formal definitions of classes, properties and individuals allows inferring new knowledge from knowledge that is already present. The fact that it is based on formal description logic makes it prone to logical reasoning and enables to infer knowledge from existing facts[1] and axioms[2].

[1] "*Fact states information about a particular individual, in the form of classes that the individual belongs to plus properties and values of that individual*" [18].

[2] "*Axioms are used to associate class and property identifiers with either partial or complete specifications of their characteristics, and to give other information about classes and properties. Axioms used to be called definitions, but they are not all definitions in the common sense of the term and thus a more neutral name has been chosen.*" [18].

The aao.owl model has been verified in the Protegè 3.4.6 using the Pellet 1.5.2 reasoner for consistency on the machine with following configuration: Processor: Intel Core i7 (2 cores 2,8 GHz each); RAM: 6 GB; Operating System: Windows 7 (64 bit). The consistency check on this machine was successful. AAO has been proven consistent.

The second important element of ontology evaluation is verification of its adherence to the problem it was designed to solve. The ontology model has been designed to support selection of the adaptation actions. In the decision algorithm there has been defined several questions that the ontology is able to give answers to. In particular, after the knowledge about the service call context is added, the AFRO adaptation ontology is queried about existence of various facts. These are the antecedents of the rules defined for AFRO. If the antecedent of a rule holds (is "true"), then the consequent must also hold. This means that for every individual that conforms to the rule, the preferred action (hasPreferredAction object property) is set. On the basis of defined rules particular adaptation actions are set. They are used by the AFRO Decision Algorithm to define actions that will be triggered by the Proxy. The actions are also dependent on the currently observed network link performance.

6.0 SUMMARY

This article presents semantic description of the service call context defined for the purpose of the Adaptation Framework For Web Services Provision (AFRO). AFRO defines a mechanism for effective web services invocation in tactical networks that are considered disadvantaged in terms of available throughput, delay and error rate. Its implementation, in the form of AFRO Proxy, performs so called adaptation actions, which are modifications of the SOAP XML messages by changing their encoding to more efficient or cutting out information that are accepted to be removed by the service requester. With these actions, the sizes of messages are significantly diminished making them better tailored to the tactical networks.

AFRO provides dynamic selection of adaptation actions to be triggered by the Proxy on the basis of user preferences and his terminal limitations. It takes advantage of the fact that limited capabilities of the user's device makes it impossible to receive by or process some pieces of data. It is therefore necessary to conserve network resources not sending the data that is not going to be consumed by the user.

The AFRO Adaptation Ontology (AAO) semantically models the user preferences in terms of the adaptation actions, his device possibilities and limitations as well as current network connection performance. This allows to unambiguously describe the service call context and, on the basis of ontology rules, provide adaptation actions tailored to particular user, his device and current network performance.

It is assumed that the AFRO Proxy will cooperate with an external element that is able to assess the current network performance on the link that the user employs to invoke web service. This link will be the bottleneck for communications with the user. On the basis of information about currently available throughput, delay and packet error rate on that link the proposed in the Article network state classification algorithm classifies the network as very good, constrained or degraded.

The proposed AFRO architecture can be exemplified by the implementation in the web service Proxy that has a web service interface and follows SOA fundamental assumption of loosely – coupled entities. The user can therefore discover the existence of the AFRO Proxy in the service registry, and subscribe to it, when necessary. Additionally, the modular architecture of the Proxy enables it to be enhanced with service orchestration module that, instead of making use of built-in adaptation engine, would search for services that are able to perform particular adaptation actions.

AFRO adaptation ontology (AAO) defines all entities that are necessary to take appropriate adaptation decision and enables to automatically select appropriate adaptation actions. Its scope covers the required level of detail in describing the entities and relationships among them. Additionally, the set of rules reflects all basic terminal characteristics that may influence usability of messages delivered to the user. It is used to build Knowledge Base about the users, their terminals and networks they are connected by in order to dynamically and automatically define the actions that the AFRO Proxy needs to take in particular network performance conditions. For the purpose of network state classification there has been proposed an algorithm that bases on information that would be received from an external Network Monitoring Element.

There has been proven that the proposed AAO model is semantically and syntactically correct and consistent. Reasoning over it provides the possibility to support the adaptation actions decisions taking into account the user preferences deriving from his role and limitations of his terminal. The SWRL rules defined for AFRO strongly support the automatic process of defining the preferred actions.

The proposed adaptation mechanism gives promising effects for low level commanders located at the battlefield, which can be supplied with information generally available on high command levels which, up to now, were very rarely distributed to tactical networks.

7.0 REFERENCES

[1] J. A. Estefan, K. Laskey, F. G. McCabe, D. Thornton, Reference Architecture Foundation for Service Oriented Architecture, Version 1.0, OASIS Committee Draft 02, 14 October 2009

[2] M. Booth, T. Buckman et al, NATO Network Enabled Feasibility Study Volume II: Detailed Report Covering a Strategy and Roadmap for Realizing an NNEC Networking and Information Infrastructure, version 2.0", NATO C3 Agency, June 2005

[3] NATO Architecture Framework (NAF) version 3.0, June 2007

[4] NATO C3 System Interoperability Directive (NID), AC/322-D(2004)0001(INV) Annex C

[5] J.Busch, An investigation into deploying web services, TN – 1229, NC3A, December 2007.

[6] R.Faucher, R. Ladysz, D.Miller, S.Musman, S.Raparla, D.Smith, Guidance on Proxy Servers for the Tactical Edge, The MITRE Corporation, MITRE TECHNICAL REPORT No. 060175, September 2006

[7] M. Weiser, The computer for the 21st century, Scientific American, (1991), 265(3):94–104

[8] Sousa, J. P. and Garlan, D. (2002). Aura: An architectural framework for user mobility in ubiquitous computing environments, In Proceedings of 3rd IEEE/IFIP Conference on Software Architecture

[9] P. F. Patel-Schneider, I. Horrocks, OWL Web Ontology Language Semantics and Abstract Syntax Section 2. Abstract Syntax, http://www.w3.org/TR/owl-semantics/syntax.html

[10] Strang et al., A context modelling survey, Workshop On Advanced Context Modelling, Reasoning And Management Associated With The Sixth International Conference On Ubiquitous Computing (Ubicomp4), Nottingham/UK

[11] I. Horrocks et al., SWRL: A Semantic Web Rule Language. Combining OWL and RuleML, W3C Member Submission 21 May 2004, http://www.w3.org/Submission/SWRL/

[12] O. Lassila, R. R. Swic, Resource Description Framework (RDF) Model and Syntax Specification, W3C Proposed Recommendation 05 January 1999, http://www.w3.org/TR/PR-rdf-syntax

[13] N. Freed, N. Borenstein, RFC2046 Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types, http://www.ietf.org/rfc/rfc2046.txt?number=2046

[14] J. Sliwa, K. Gleba, W. Chmiel, P. Szwed, A. Glowacz, IOEM- ontology engineering methodology for large systems, Lecture Notes in Computer Science, volume 6922

[15] C. Kiss, Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 2.0, W3C Working Draft 30 April 2007, http://www.w3.org/TR/2007/WD-CCPP-struct-vocab2-20070430/

[16] J.Śliwa, T.Podlasek, M.Amanowicz, Web Services Efficiency in Disadvantaged Environment, JTIT 2010

[17] A. Tarski, Introduction to Logic and to the Methodology of Deductive Sciences, Second Edition, Dover Publications, Inc., New York 1946, ISBN 0-486-28462-X

[18] P. F. Patel-Schneider, I. Horrocks, OWL Web Ontology Language Semantics and Abstract Syntax Section 2. Abstract Syntax, http://www.w3.org/TR/owl-semantics/syntax.html