Access Control for Social Care Platforms Using Fast Healthcare Interoperability Resources

Marco Rosa[1], João Paulo Barraca[1][0000-0002-5029-6191] and Nelson Pacheco Rocha[1][0000-0002-3801-7249]

[1] Universidade de Aveiro, Campus Universitário de Santiago, Aveiro, Portugal
{marcofrosa, jpbarraca, npr}@ua.pt

Abstract. The definition of authorization policies is essential to prevent information misuse and to guarantee that only authorized personnel can access specific information. Since not everyone is familiar with special purpose languages, an interpretation tool can allow the management of policies and rules using natural languages. This paper focuses on a parser developed as a component of a platform to support the care of community-dwelling older adults, the SOCIAL platform, allowing to create, read, update and delete authorization policies and rules, using natural languages.

Keywords: ABAC, XACML, access control, parser, natural language, FHIR.

1 Introduction

Health care and social care systems and platforms need to handle large quantities of care receiver's information. Since these systems and platforms manage sensitive personal information, including clinical parameters, a major concern is related to the authorization policies, especially because interoperability with other systems and platforms must be achieved.

Therefore, authorization mechanisms to decide if someone can access certain resources are crucial to promote privacy and confidentiality by preventing illegal accesses. However, the definition of authorization policies using special purpose policy languages may be hard to interpret and modify to those with a low technical knowledge whenever the access rules need to be updated. This means that specialized support are required to understand policy languages and to apply them, which implies additional resources.

The present paper purposes a natural language and policy language parser that can help anyone, even with no knowledge about specific policy languages, to create, read, update and delete rules related to authorization policies to be integrated with the Fast Healthcare Interoperability Resources (FHIR) [1], developed by Health Level Seven International (HL7), to promote interoperability across applications and systems.

The following sections will explain available solutions for access control, the target platform for the proposed solution, the SOCIAL platform [2], and why an access control is important to fulfill its goals, the requirements for a natural language and policy language parser, its implementation and the evaluation to verify if the defined requirements are met.

2 Related Work

There are several open platforms [3], such as the Substitutable Medical Applications, Reusable Technologies (SMART) [4, 5] aiming to provide agile developments of new applications targeting electronic health records. Other platforms consider the specificities of the information required for the social care provision (i.e. electronic social records [6]), such as Senior Care Connect [7] and Ankira [8], aiming to deliver services to provide better quality of life for senior citizens, namely by promoting their empowerment and the empowerment of their formal and informal care, and assistance providers.

Both healthcare and social care platforms benefit with the integration with other systems and platforms, which means that interoperability standards assume a paramount importance to allow the exchange of the care receiver's information. In this respect, the FHIR has become increasingly important to provide longitudinal records of patient's health and healthcare [9] because it combines the best features of other known specifications, such as HL7 v2, HL7 v3, and Clinical Document. FHIR is being integrated in different systems and platforms, such as System C [10], SocialCare [11] or SOCIAL [2].

To avoid unauthorized accesses health care and social care platforms it is essential to develop access control mechanisms, such as Attribute-based Access Control (ABAC) or Role-based Access Control (RBAC).

The eXtensible Access Control Markup Language (XACML) [12], whose structure is a combination between Extensible Markup Language (XML) and Security Assertion Markup Language (SAML) [13], is a standard that allows handling requests, regarding data access, using an ABAC policy language. These requests related to the access of specific resources are be analyzed through a set of pre-established rules (of a certain policy) so that a decision can be achieved.

By using XACML, each rule can be divided into different categories, each one with different attributes, which can have several data types, namely double, boolean, string or date.

The decision for the access request is obtained after comparing all the values of the attributes of the respective categories, between the request and the previously defined rules. Depending on how a specific decision rule was defined, a request result could be a denial or a permit (the access). With XACML it is also possible to establish comparisons, either by verifying if two values are equal, greater or less or by using "and" or "or" operators between the different attributes of a rule, to reach a decision. Additionally, it is also possible to return more information than the decision result, such as the element "AdviceExpressions" that can specify additional restrictions regarding access decisions.

With XACML, policies and their respective rules can have different levels of detail, according to the different number of attributes needed before reaching a decision. This feature makes this standard a perfect fit for systems that handle great quantities of information, with different types of data.

Several studies highlight how XACML can help a system with sensitive information (e.g. health-related information [14, 15]) and there are already examples of successful integration with existent systems [16], which proves its usefulness and flexibility. Because of this, tools were developed to help the management of access policies defined using XACML. Turner [17] developed a XACML parser and Stepien and colleagues [18] proposed a solution that can allow people with low technical understanding of XACML (e.g. clinicians working in the emergency, military or medical field) to create access rules.

However, those solutions are focused on healthcare, and do not consider the specificities of social care and informal care. Furthermore, these solutions don't take advantage of the "AdviceExpressions" XACML element, which is very dependent on the objectives and goals of the system being implemented. This turns even more important the development of custom XACML parsers, so that a system or platform can be able to adapt to any kind of scenario, regarding the access of its information.

Moreover, to the best of our knowledge, no previous studies have reported the development of a natural language and policy language parser to manage authorization policies considering the specificities of FHIR.

3 Profile Authorization for the SOCIAL platform

The SOCIAL platform [2] aims to surpass the so-called health and social care divide [6] and to provide information services to support the care and assistance of community-dwelling older adults. The fundamental technological element is a platform of services, which include all the structural components that are required to support a range of coherent applications to ensure an integrated, consistent and cross-cutting view of the information of the care receivers, and efficient communication with and among caregivers [19].

In terms of architecture, the SOCIAL platform allows the integration of different applications by using FHIR. According to FHIR, data are represented as resources, which are JSON (or XML) objects with healthcare related fields and values (e.g. Patient, Practitioner, DiagnosticReport, Coverage, Questionnaire or Consent). In this respect, care receivers of the SOCIAL platform can be referenced as a Patient resource, where their demographic and other administrative information are structured as attributes (e.g. name or address). Other relevant FHIR resources that can be integrated with the SOCIAL platform are Practitioner (for the care or assistance provider), PractitionerRole (for the care or assistance provider role), Organization, CarePlan and Task.

The FHIR resources may have security labels for three different purposes: to indicate the permissions related to information operations, such as read or modify, to indicate what resources can be returned, and to indicate how specific information should be dealt with. Requests can also add their own security labels (e.g. break-the-glass protocol to allow a physician dealing with an emergency situation to access information of patient even when not having access permissions). The security labels can have different categorizations, such as confidentiality, sensitivity, compartment, integrity or handling.

Integrating ABAC mechanisms with this standard is possible, since the FHIR resources are also defined by their attributes.

3.1 Requirements

The main goal was to develop a parser for the SOCIAL platform compatible with FHIR resources. The parser should be versatile, so that any qualified user (whether it be an administrator, a case manager or a care provider) can manage the access rules of any exchanged information inside or outside the platform:

Flexibility in policy management - Anyone with the needed qualifications should be able to manage the access policies for their applications or services, regardless of their technical knowledge. A user of an integrated application or service should not need to know the policy language structure or ask for help to someone who knows, to create, read, update and delete access rules. This should be able to be done using a simple language that is understandable by most.

Moreover, the following requirements were also considered:

Attribute-based policies and rules - **s**ince the SOCIAL platform handles sensitive data, the access of these data should not be too generic. Using a single identifying attribute (e.g. the user's role) to decide if someone should access a resource can create scenarios of misuse of information. So, the SOCIAL platform access control should use and analyze several different attributes before reaching a decision, so that users with certain identifying attributes can access only the data to which they are authorized.

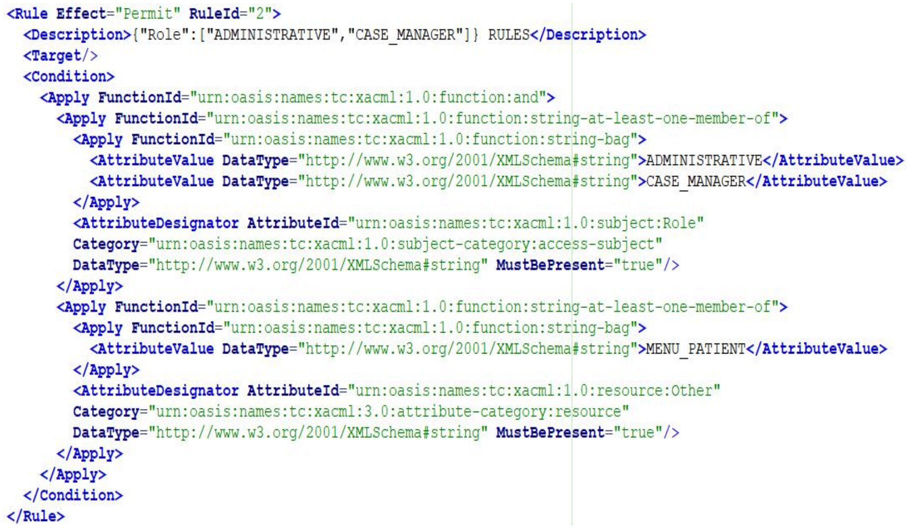
Simple decision responses, with possible additional information - The responses regarding an access request should be simple enough, so that the requester can easily understand its meaning. However, in some cases, there can be some restrictions that must be enforced by the access control component or that should be notified to the requester, so that it can act accordingly.

Detailed rules, for the most specific use cases - there can be several conditions for a specific access, and it should be possible to create a rule that can fulfill those requirements, regardless of their complexity. Those restrictions may be about the subject, the resource, the action, other categories, or even all of them. The access control should be able of handling any kind of situation.

FHIR resources integration - since the FHIR resources have several attributes, these can be used to manage the policies and rules of a system, by using their respective values. Using these attributes plus the platform-specific attributes (e.g. IDs, roles or list of care receivers) allows the handling of different types of access scenarios, which can fulfill the requirements of the SOCIAL platform and respective applications and services.

3.2 XACML Parsing

XACML defines categories and attributes to distinguish the different entities involved in an access request. By default, there are four different categories: "access-subject" is the entity that is requesting the access, "resource" is what is going to be accessed, "action" is the type of access and "environment" is additional information regarding the access. Each one of these categories can have several attributes.



```
<Rule Effect="Permit" RuleId="2">
  <Description>{"Role":["ADMINISTRATIVE","CASE_MANAGER"]} RULES</Description>
  <Target/>
  <Condition>
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">ADMINISTRATIVE</AttributeValue>
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">CASE_MANAGER</AttributeValue>
        </Apply>
        <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:Role"
        Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
        DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
      </Apply>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">MENU_PATIENT</AttributeValue>
        </Apply>
        <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Other"
        Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
        DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
      </Apply>
    </Apply>
  </Condition>
</Rule>
```

F

ig. 1. XACML structure example.

Considering the structure of XACML (see Fig. 1), its understanding can be difficult to most. However, managing XACML rules should not be dependent on the understanding of this language, and anyone with the proper authority should be able to create, read, update and delete policy rules. Developing a parser that can translate both from XACML to natural language and natural language to XACML can prove to be useful in maintaining this type of access control.

The rule (in natural language) might have the following structure:

If a "<SUBJECT_ATTRIBUTE>" is "<SUBJECT_VALUE>",
it can "<ACTION>" the "<ENVIRONMENT>"
if the "<RESOURCE_ATTRIBUTE>" of
the resource is "<RESOURCE_VALUE>"

An example of the application of this rule could be:

If a "role" is "case manager",
it can "read" the "profile"
if the "role" of the resource" is "older adult"

Therefore, being established a structure for the definition of the rules, it is possible to parse the rules to their respective XACML format. Moreover, the inverse (i.e. the translation from XACML to natural language) is also possible. For instance, from the excerpt of XACML code presented in the Table 1 is possible to retrieve the following rule:

If a "role" is "administrative" or is "case manager",

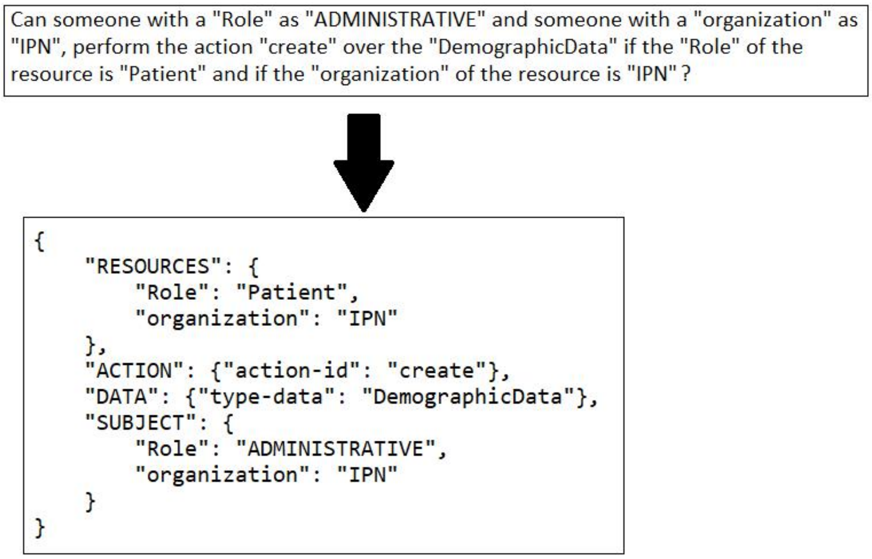
it can "create" and "update" and "read" the "address",
the “name” and the “telecom”
    if the "role" of the resource is "patient".
    Advices: "SAME_ORGANIZATION"

Table 1. Category, AttributeId and respective value of XACML rule.

Category	AttributeId	Value
(…):xacml:1.0:subject-category:access-subject	(…):xacml:1.0:subject:role	"administrative" or "case manager"
(…):xacml:3.0:attribute-category:action	(…):xacml:1.0:action:action-id	"create" or "update" or "read"
(…):xacml:3.0:attribute-category:environment	(…):xacml:1.0:environment:type-data	"address" or "name" or "telecom"
(…):xacml:3.0:attribute-category:resource	(…):xacml:1.0:resource:role	"patient"

The access requests themselves can also be sent in a natural language structure, from the user’s perspective and then be processed accordingly by the parser. For example:

Can someone with a "<SUBJECT_ATTRIBUTE>" as
"<SUBJECT_VALUE>",
    perform the action "<ACTION>" over the
    "<ENVIRONMENT>"
        if the "<RESOURCE_ATTRIBUTE>"
        of the resource is "<RESOURCE_VALUE>"?



Can someone with a "Role" as "ADMINISTRATIVE" and someone with a "organization" as "IPN", perform the action "create" over the "DemographicData" if the "Role" of the resource is "Patient" and if the "organization" of the resource is "IPN" ?

```
{
    "RESOURCES": {
        "Role": "Patient",
        "organization": "IPN"
    },
    "ACTION": {"action-id": "create"},
    "DATA": {"type-data": "DemographicData"},
    "SUBJECT": {
        "Role": "ADMINISTRATIVE",
        "organization": "IPN"
    }
}
```

Fig. 2. Parsing from natural language to XACML.

The parser should then process this phrase and transform it into a JSON Object (see Fig. 2). Considering, the possibility to add the "AdviceExpressions" element using the parser, additional restrictions can be defined. For example, for scenarios where the access decision is "Permit", but only if a certain attribute value is the same between the subject and resource, we can add, after an access statement, for instance to associated to a code, that is understood by the access control component:

> However, the "subject" "<SUBJECT_ATTRIBUTE>"
> must be the same as the "resource"
> "<RESOURCE_ATTRIBUTE>"

With the addition of "AdviceExpressions" several possibilities can be considered, such as restrictions to certain environment fields or accesses during specific periods of time. Therefore, additional steps must be considered before enforcing a final decision.

4 Evaluation

After the implementation of the parser using the JAVA language, an evaluation was performed. For the evaluation it was considered a scenario where it is required the possibility of sending access requests that should be verified if a set of rules was followed or not, by using both the SOCIAL platform identifying attributes (e.g. the “role” of the users in the platform) and the FHIR resources attributes (e.g. the attributes of the Patient FHIR Resource such as “name” or “address”).

First, three distinct rules were created, using the natural language parser:

- Rule 1 - If a "role" is "case manager", it can "update" the "address" if the "role" of the resource is "patient".
- Rule 2 - If a "role" is "care receiver", it can "read", "update" and "create" the "photo" if the "role" of the resource is "patient". However, the "subject" "organization" must be the same as the "resource" "organization".
- Rule 3 - If an "organization" is "City Council A", it can "read" the "name" and the "address" if the "organization" of the resource is "City Council B".

Next, access requests were sent and it was verified if the rules were processed correctly (also using the natural language parser):

- Question 1 - Can someone with a "role" as "case manager", perform the action "delete" over the "address" if the "role" of the resource is "patient"?
- Question 2 - Can someone with a "role" as "case manager", perform the action "update" over the "address " if the "role" of the resource is "patient"?
- Question 3 - Can someone with a "role" as "care receiver" and with an "organization" as "Retirement Home A", perform the action

"update" over the "photo" if the "role" of the resource is "patient" and if the "organization" of the resource is "Retirement Home A"?

- Question 4 - Can someone with a "role" as "care receiver" and with an "organization" as "Retirement Home B", perform the action "update" over the "photo" if the "role" of the resource is "patient" and if the "organization" of the resource is "Retirement Home A"?
- Question 5 - Can someone with a "role" as "care receiver" and with an "organization" as "Retirement Home A" perform the action "update" over the "photo" if the "role" of the resource is "patient" and if the "organization" of the resource is "Retirement Home B"?
- Question 6 - Can someone with an "organization" as "City Council A", perform the action "read" over the "address" if the "organization" of the resource is "City Council B"?
- Question 7 - Can someone with an "organization" as "City Council A", perform the action "read" over the "telecom" if the "organization" of the resource is "City Council B"?

Question 1 and 2 are for testing rule 1, while question 3, 4 and 5 are for testing rule 2 and finally, question 6 and 7 are for testing rule 3.

The seven questions are represented in Table 2 (for the attributes of the access-subject and resource category) and Table 3 (for the attributes of the action and environment category and access decision result).

Table 2. Access Request tests (access-subject and resource).

Question	Access-subject	Resource
1	role:*case manager*	role:*patient*
2	role:*case manager*	role: *patient*
3	role:*care receiver*, organization:*Retirement Home A*	role: *patient*, organization:*Retirement Home A*
4	role:*care receiver*, organization:*Retirement Home B*	role:*patient*, organization:*Retirement Home A*
5	role:*care receiver*, organization:*Retirement Home A*	role:*patient*, organization:*Retirement Home B*
6	organization:*City Council A*	organization:*City Council B*
7	organization:*City Council A*	organization:*City Council B*

As we can see in Table 3, the first question decision is denied, since the "case manager" can only "update" the "address" of the "patient". The fourth and fifth question also got a denial to its access request because only "care receiver" from "Retirement Home A" can "update" the "photo" of the "patient" from "Retirement Home A". The seventh question got a denial answer because the "City Council A" subjects can only "read" the "name" and "address" from resources from "City Council B".

Table 3. Access Request tests (action, environment and decision).

Question	Action	Environment	Decision
1	Delete	address	Deny

2	Update	address	Permit
3	Update	photo	Permit
4	Update	photo	Deny
5	Update	photo	Deny
6	Read	address	Permit
7	Read	telecom	Deny

5 Conclusion

Using a natural language parser not only can help someone with low understanding of the XACML standard to be able to create and verify XACML rules. The study reported in the present paper shows that the parser do not disrupt the XACML structure and can be applied to FHIR resources (as proven by the decision answers obtained).

The parser can specify different attributes for any category and adapt them to create detailed access rules, which can be suitable to almost any kind of scenario. Moreover, by using the "AdviceExpressions" element, it can even add another layer of complexity over a certain rule and make the access over a certain resource even more restrictive.

Using XACML in a system's access control, that handles several distinct data, allows for a more secure system that can satisfy any requirement imposed by any institution or organization, regarding their data access. Therefore, developing a parser that can translate XACML to natural language, allowing anyone to directly manage their respective policies, might help an effective use of XACML.

Acknowledgements

This work was supported by Sistema de Incentivos à Investigação e Desenvolvimento Tecnológico (SI I&DT) of the Programa Portugal 2020, through Programa Operacional Competitividade e Internacionalização and/or Programa Operacional do Centro do FEDER - Fundo Europeu de Desenvolvimento Regional, under Social Cooperation for Integrated Assisted Living (SOCIAL), project number 017861.

References

1. Baines, S., Hill, P., Garrety, K.: What happens when digital information systems are brought into health and social care? Comparing approaches to social policy in England and Australia. Social Policy and Society, 13(4), 569-578 (2014).
2. Sousa, M., Arieira, L., Queirós, A., Martins, A.I., Rocha, N.P., Augusto, F., Duarte, F., Neves, T., Damasceno, A.: SOCIAL platform. In Advances in Intelligent Systems and Computing, 746, 1162-1168 (2018).
3. Defining an Open Platform. Apperta Foundation (2017).
4. Mandl, K.D., Mandel, J.C., Murphy, S.N., Bernstam, E.V., Ramoni, R.L., Kreda, D.A., et al.: The SMART Platform: Early experience enabling substitutable applications for electronic health records. J Am Med Informatics Assoc, 19(4), 597-603 (2012).

5. Chaballout, B.H., Shaw, R.J., Reuter-Rice, K.: The SMART healthcare solution. Advances in Precision Medicine, 2(1), 1-3 (2017).
6. Rigby, M.: Integrating health and social care informatics to enable holistic health care. Stud Health Technol Inform., 177, 41-51 (2012).
7. Kristal, L.: Senior Care Connect Inc. [Online]. Available: https://www.seniorcareconnect.co/ (visited on 2018/10/22).
8. Metatheke Software: Ankira. [Online]. Available: https://ankira.pt/en/platform/ (visited on 2018/10/22).
9. HL7, Fast Healthcare Interoperability Resources. [Online]. Available: https://www.hl7.org/fhir/ (visited on 2018/10/22).
10. Hoeksma, J.: System C commits to ‘full FHIR support’ to drive interoperability, September 2018. [Online]. Available: https://www.digitalhealth.net/2018/09/system-c-commits-to-full-fhir-support-to-drive-interoperability/ (visited on 2018/10/22).
11. Chu, D.: SocialCare. [Online]. Available: https://www.socialcare.com/ (visited on 2018/10/22).
12. OASIS, XACML, Jan. 2013. [Online]. Available: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html (visited on 2018/10/22).
13. OASIS, Security Assertion Markup Language. [Online]. Available: https://wiki.oasis-open.org/security/FrontPage (visited on 2018/10/22).
14. Vora, J. et al.: Ensuring Privacy and Security in E- Health Records. In International Conference on Computer, Information and Telecommunication Systems, pp. 1-5, IEEE, Colmar, France (2018).
15. Ray, I. et al.: Applying attribute based access control for privacy preserving health data disclosure. IEEE-EMBS International Conference on Biomedical and Health Informatics, pp. 1-4, IEEE, Las Vegas, Nevada, USA (2016).
16. Atiq, A.M., Alsulaiman, L.A.: Using XACML to enhance compliance with privacy regulations in health sector. In 2016 World Symposium on Computer Applications & Research (WSCAR), pp. 53-58, IEEE, Cairo, Egypt (2016).
17. Turner, R.C.: Proposed Model for Natural Language ABAC Authoring. In Proceedings of the 2nd ACM Workshop on Attribute-Based Access Control - ABAC ’17, pp. 61-72, ACM, Scottsdale, Arizona, USA (2017)
18. Stepien, B. et al.: A non-technical XACML target editor for dynamic access control systems. In 2014 International Conference on Collaboration Technologies and Systems, pp. 150-157, IEEE, Minneapolis, Minnesota, USA (2014).
19. Santana, S., Dias, A., Souza, E., Rocha, N.: The Domiciliary Support Service in Portugal and the change of paradigm in care provision. International journal of integrated care, 7(1) (2007).