Applying the US2StarTool in a real example

Renato Mesquita[1], Aline Jaqueira[1], Márcia Lucena[1],
Celso Agra[2] and Fernanda Alencar[23]

[1]Departamento de Informática e Matemática Aplicada – UFRN
{rmsnatal,alineopj}@gmail.com; marciaj@dimap.ufrn.br
[2]Programa de Pós-Graduação em Engenharia da Computação – UPE
[3]Departamento de Eletrônica e Sistemas - UFPE
celsoagra@gmail.com; fernanda.ralencar@ufpe.br

1   Introduction

To illustrate the use of US2StarTool were used 19 user stories relating to the requirements of the BR-PlantHistorian and BR-PlantViewer systems maintained by Logap company. The Logap is a company incubated at Inova Metropolis, within the Federal University of Rio Grande do Norte (UFRN), which operates in the information technology area.

The BR-PlantHistorian is a system developed by UFRN Industrial Computer Lab for information management of the variables of industrial processes optimally with proper techniques for big data management. The aim is to assist the management of processes, enabling store and distribute the data for each plant.

The BR-PlantViewer system was developed by UFRN in partnership with Petrobras, to monitor the variables of industrial processes via web browsers or mobile devices. Its purpose is to assist in managing processes, enabling the monitoring of data every industrial plant in real time or from historical data through screens and dynamically built graphs. Currently, the system can obtain information from the process variables monitored by the PI System, which is a widely used system for managing information processes, BR-PlantHistorian and directly from BR-Collector.

2   User Stories of systems BR-PlantHistorian and BR-PlantViewer

The table 1 shows the user stories relating to the development of BR-PlantHistorian and BR-PlantViewer systems.

Table 1. User Stories of BR-PlantHistorian and BR-PlantViewer.

User Story
Como usuário BR-PlantHistorian Eu quero alterar cor tela login Para que diferenciar do BR-PlantViewer
Como admin Eu quero organizar hierarquia de áreas Para que facilitar criar arquitetura de áreas do processo
Como gestor do BR-Historian Eu quero cadastro de tags dividido em passos Para que facilitar cadastro de tags
Como gestor do BR-Historian Eu quero cadastro de tags organizado em abas Para que facilitar cadastro de tags
Como usuário BR-PlantViewer Eu quero reformulação componente de vetores Para que escolher dados em X e Y na criação de gráficos

Como usuário BR-PlantViewer **Eu quero** componente que represente gráfico pizza **Para que** facilitar visualização de dados
Como gestor do BR-Historian **Eu quero** alterar parâmetro do sist. de forma simples **Para que** facilitar gestão de mudanças
Como usuário BR-PlantViewer **Eu quero** nomear eixo Y **Para que** deixar explícito variável contabilizada
Como engenheiro **Eu quero** criar tag no BR-PlantHistorian **Para que** maior integração com outras apps
Como usuário BR-PlantViewer **Eu quero** ter componente gráfico **Para que** facilitar visualização de dados
Como gestor do BR-Historian **Eu quero** visualizar arquivos de backup **Para que** validar geração de backups
Como usuário BR-PlantViewer **Eu quero** criar meu próprio componente **Para que** reaproveitar representatividade visual
Como usuário BR-PlantViewer **Eu quero** renomear gráfico de controle preditivo para gráfico de vetores **Para que** nomenclatura seja mais abrangente
Como usuário BR-PlantViewer **Eu quero** nomear visualização da tag **Para que** ajustar apresentação do gráfico
Como engenheiro **Eu quero** armazenar variáveis do tipo array **Para que** facilitar armazenar informações que precisem de múltiplos pontos relacionados a um mesmo timestamp
Como usuário BR-PlantViewer **Eu quero** renomear categoria e legenda **Para que** facilitar manutenção dos gráficos barra
Como admin **Eu quero** controle de permissão de acesso às tags **Para que** limitar o acesso a um grupo de tags
Como integrador de sistemas **Eu quero** função p/ converter tag em id **Para que** realizar chamadas às outras funções que dependem de um id
Como integrador de sistemas **Eu quero** Historian-Collector tenha uma interface **Para que** receber dados de outras apps através de uma api

3   XMI file obtained by US2StarTool

After performing the mapping of user stories to i* model from the US2StarTool, it is possible to export to a file, in XMI format, able to be read by IstarTool. The IstarTool is responsible for generating the i* model graphic. Below we can see the file XMI code generated by US2StarTool.

```xml
<?xml version="1.0" encoding="UTF-8"?>
```

```xml
<istar:Model xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
xmlns:istar="http://www.cin.ufpe.br/istar" title="Logap Example">
  <elements name="diferenciar do BR-PlantViewer" type="GOAL"/>
  <elements name="facilitar criar arquitetura de áreas do processo"
type="GOAL"/>
  <elements name="facilitar cadastro de tags" type="GOAL"/>
  <elements name="escolher dados em X e Y na criação de gráficos "
type="GOAL"/>
  <elements name="facilitar visualização de dados" type="GOAL"/>
  <elements name="facilitar gestão de mudanças" type="GOAL"/>
  <elements name="deixar explícito variável contabilizada" type="GOAL"/>
  <elements name="maior integração com outras apps" type="GOAL"/>
  <elements name="validar geração de backups" type="GOAL"/>
  <elements name="reaproveitar representatividade visual" type="GOAL"/>
  <elements name="nomenclatura seja mais abrangente" type="GOAL"/>
  <elements name="ajustar apresentação do gráfico" type="GOAL"/>
  <elements name="facilitar armazenar informações que precisem de múltiplos
pontos relacionados a um mesmo timestamp" type="GOAL"/>
  <elements name="facilitar manutenção dos gráficos barra" type="GOAL"/>
  <elements name="limitar o acesso a um grupo de tags" type="GOAL"/>
  <elements name="realizar chamadas às outras funções que dependem de um id "
type="GOAL"/>
  <elements name="receber dados de outras apps através de uma api "
type="GOAL"/>
  <actors name="usuário BR-PlantHistorian" type="ACTOR"/>
  <actors name="admin" type="ACTOR"/>
  <actors name="gestor do BR-Historian " type="ACTOR"/>
  <actors name="usuário BR-PlantViewer " type="ACTOR"/>
  <actors name="engenheiro" type="ACTOR"/>
  <actors name="integrador de sistemas" type="ACTOR"/>
  <actors name="SystemActor" type="ACTORBOUNDARY">
    <decompositionsTask source="//@actors.6/@elements.18"
target="//@actors.6/@elements.20"/>
    <decompositionsTask source="//@actors.6/@elements.18"
target="//@actors.6/@elements.2"/>
    <decompositionsTask source="//@actors.6/@elements.19"
target="//@actors.6/@elements.4"/>
    <decompositionsTask source="//@actors.6/@elements.19"
target="//@actors.6/@elements.8"/>
    <elements name="alterar cor tela login" type="TASK"/>
    <elements name="organizar hierarquia de áreas" type="TASK"/>
    <elements name="cadastro de tags organizado em abas" type="TASK"/>
    <elements name="reformulação componente de vetores" type="TASK"/>
    <elements name="componente que represente gráfico pizza" type="TASK"/>
    <elements name="alterar parâmetro do sist. de forma simples" type="TASK"/>
    <elements name="nomear eixo Y" type="TASK"/>
    <elements name="criar tag no BR-PlantHistorian " type="TASK"/>
    <elements name="ter componente gráfico" type="TASK"/>
    <elements name="visualizar arquivos de backup" type="TASK"/>
    <elements name="criar meu próprio componente" type="TASK"/>
    <elements name="renomear gráfico de controle preditivo para gráfico de
vetores " type="TASK"/>
    <elements name="nomear visualização da tag" type="TASK"/>
    <elements name="armazenar variáveis do tipo array" type="TASK"/>
    <elements name="renomear categoria e legenda" type="TASK"/>
    <elements name="controle de permissão de acesso às tags" type="TASK"/>
    <elements name="função p/ converter tag em id" type="TASK"/>
    <elements name="Historian-collector tenha uma interface" type="TASK"/>
    <elements name="{Generic TASK by goal: facilitar cadastro de tags}"
type="TASK"/>
    <elements name="{Generic TASK by goal: facilitar visualização de dados}"
type="TASK"/>
    <elements name="cadastro de tags dividido em passos" type="TASK"/>
  </actors>
  <links source="//@actors.0" target="//@elements.0" name="" type="COMMITED"/>
  <links source="//@actors.1" target="//@elements.1" name="" type="COMMITED"/>
  <links source="//@actors.2" target="//@elements.2" name="" type="COMMITED"/>
  <links source="//@actors.3" target="//@elements.3" name="" type="COMMITED"/>
```

```
  <links source="//@actors.3" target="//@elements.4" name="" type="COMMITED"/>
  <links source="//@actors.2" target="//@elements.5" name="" type="COMMITED"/>
  <links source="//@actors.3" target="//@elements.6" name="" type="COMMITED"/>
  <links source="//@actors.4" target="//@elements.7" name="" type="COMMITED"/>
  <links source="//@actors.2" target="//@elements.8" name="" type="COMMITED"/>
  <links source="//@actors.3" target="//@elements.9" name="" type="COMMITED"/>
  <links source="//@actors.3" target="//@elements.10" name=""
type="COMMITED"/>
  <links source="//@actors.3" target="//@elements.11" name=""
type="COMMITED"/>
  <links source="//@actors.4" target="//@elements.12" name=""
type="COMMITED"/>
  <links source="//@actors.3" target="//@elements.13" name=""
type="COMMITED"/>
  <links source="//@actors.1" target="//@elements.14" name=""
type="COMMITED"/>
  <links source="//@actors.5" target="//@elements.15" name=""
type="COMMITED"/>
  <links source="//@actors.5" target="//@elements.16" name=""
type="COMMITED"/>
  <links source="//@elements.0" target="//@actors.6/@elements.0" name=""
type="COMMITED"/>
  <links source="//@elements.1" target="//@actors.6/@elements.1" name=""
type="COMMITED"/>
  <links source="//@elements.3" target="//@actors.6/@elements.3" name=""
type="COMMITED"/>
  <links source="//@elements.5" target="//@actors.6/@elements.5" name=""
type="COMMITED"/>
  <links source="//@elements.6" target="//@actors.6/@elements.6" name=""
type="COMMITED"/>
  <links source="//@elements.7" target="//@actors.6/@elements.7" name=""
type="COMMITED"/>
  <links source="//@elements.8" target="//@actors.6/@elements.9" name=""
type="COMMITED"/>
  <links source="//@elements.9" target="//@actors.6/@elements.10" name=""
type="COMMITED"/>
  <links source="//@elements.10" target="//@actors.6/@elements.11" name=""
type="COMMITED"/>
  <links source="//@elements.11" target="//@actors.6/@elements.12" name=""
type="COMMITED"/>
  <links source="//@elements.12" target="//@actors.6/@elements.13" name=""
type="COMMITED"/>
  <links source="//@elements.13" target="//@actors.6/@elements.14" name=""
type="COMMITED"/>
  <links source="//@elements.14" target="//@actors.6/@elements.15" name=""
type="COMMITED"/>
  <links source="//@elements.15" target="//@actors.6/@elements.16" name=""
type="COMMITED"/>
  <links source="//@elements.16" target="//@actors.6/@elements.17" name=""
type="COMMITED"/>
  <links source="//@elements.2" target="//@actors.6/@elements.18" name=""
type="COMMITED"/>
  <links source="//@elements.4" target="//@actors.6/@elements.19" name=""
type="COMMITED"/>
</istar:Model>
```

4   Graphic of i* model generated by IstarTool from the output file from US2StarTool
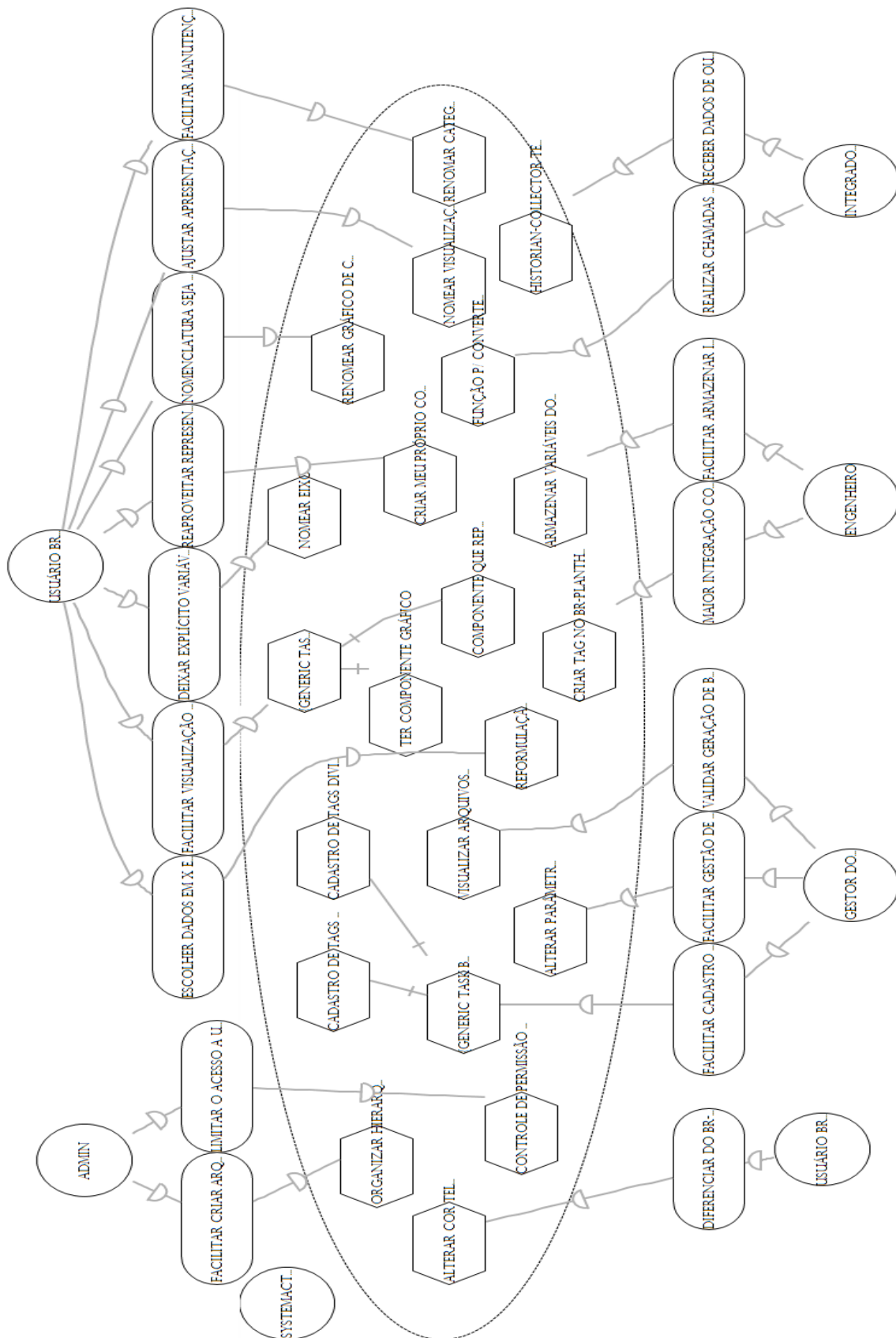
5   Analysis of results

With the graphical representation of the i* model generated by IstarTool and based on the XMI file that has been exported by US2StarTool, we can establish some conclusions about what was obtained. From the diagram i * model was possible:

- Get a graphical representation of system requirements;
  - o From the reading of user stories it is difficult to have a general understanding of the system, even with a non-significant amount of user stories. Now, with the graphical representation of the i* model, we can have a better view of the actors and their dependencies.
  - o The non-repetition of actors helped reduce the complexity of the diagram. In the case of the actor "BR-PlantViwer User" in user stories model was repeated 7 times, but viewing the i* model it is represented only one time.
- Facilitate the viewing of the role of an actor within the system;
  - o By not allowing the repetition of actors, it became easier for all the dependencies that an actor has with your goals. Thus, it facilitates the understanding of how the actor participates in the system;
- Show how the actors reach their goals.
  - o With the decompositions of tasks it is possible view the tasks that an actor needs to do to reach a certain goal. In this example, we can see that for the actor "gestor do BR-Historian" reach the goal "facilitar cadastro de tags", it needs to perform the tasks "ter cadastro de tags organizado em abas" and "ter cadastro de tags dividido em passos".
- With the grouping of requirements and its graphical representation, the i* model assists in assigning steps and prioritization of tasks in the context of an agile development environment.
  - o The best understanding of the system requirements, obtained by graphical representation helps in determining the most important tasks, defining its priority.
  - o The grouping of requirements, obtained by a more simplified view of the model, with the decomposition of actors and tasks, to determine the related features and assign them to the same development team.