BILROST: Handling Actuators of the Internet of Things through Tweets on Twitter using a Domain-Specific Language

Daniel Meana-Llorián, Cristian González García*, B. Cristina Pelayo G-Bustelo, Juan Manuel Cueva Lovelle

MDE Research Group, Department of Computer Science, University of Oviedo, Oviedo, Asturias (Spain)

uniR
LA UNIVERSIDAD
EN INTERNET

Abstract

In recent years, many investigations have appeared that combine the Internet of Things and Social Networks. Some of them addressed the interconnection of objects as Social Networks interconnect people, and others addressed the connection between objects and people. However, they usually used interfaces created for that purpose instead of using familiar interfaces for users. Why not integrate Smart Objects in traditional Social Networks? Why not control Smart Objects through natural interactions in Social Networks? The goal of this paper is to make easier to create applications that allow non-experts users to control Smart Objects actuators through Social Networks through the proposal of a novel approach to connect objects and people using Social Networks. This proposal will address how to use Twitter so that objects could perform actions based on Twitter users' posts. Moreover, it will be presented a Domain-Specific language that could help in the task of defining the actions that objects could perform when people publish specific content on Twitter.

I. Introduction

THE Internet of Things (IoT) is a term that has gained popularity in recent years among common people due to the wishes for interconnecting the whole things around them. They want to connect objects located at home to the Internet like the fridge, the oven, and so on, to manage them or know real-time information about them. Some examples of their expectations can be a fridge with the capability of alerting them when a product is running out, or an oven capable of turning on when they are arriving home. Moreover, the rise of mobile devices like smartphones, tablets, wearables, or any other devices connected to the internet like sensors, smart tags, and so on, has contributed to the popularity of the IoT.

Despite the growing popularity of the IoT, it is not so present in people's lives as was expected. What are the causes? The answer could be the complexity of managing the Smart Objects. The Smart Objects are usually composed of other objects without intelligence like sensors and actuators [1], also called Not-Smart Objects. On the one hand, managing Smart Objects' actuators could be complex: what actions to do, when the actuator must work, or how it must do it. On the other side, recollecting and interpreting data from sensors could also be so complex. Moreover, establishing connections with these Smart Objects is also an advanced task. Because of that, the goal of this work is to reduce the complexity of developing a specific type

of application that Smart Objects could run. It aims to facilitate the creation of applications for users without advanced programming knowledge (hereinafter non-expert users) that allow them to control Smart Objects' actuators through a novel way of communication, the traditional Social Networks like Twitter.

The use of Social Networks in communication with the Smart Objects has several advantages over other commons solutions like architectures client-server over HTTP protocol. For instance, this approach does not require using a specific application to intercommunicate users with objects. Users could use any application that allows them to use the chosen Social Network. Besides, this solution may be more intuitive for people who use Social Networks in their day-to-day lives.

Therefore, the hypothesis is the next: **It is possible to facilitate the creation of applications that allow non-experts users to control Smart Objects' actuators through Social Networks for humans.**

To achieve this goal, it is proposed the creation of a Domain-Specific Language (DSL) applying Model-Driven Engineering (MDE), that have been called Bilrost Specific Language (BSL). The BSL was designed focused on the ease of use and it provides the required features to enable users to define the rules and properties needed to set up the Smart Objects' actuators with their actions and to communicate Smart Objects and users through Social Networks. This proposal is capable of generating application projects for Smart Objects where the whole logic needed to communicate the Smart Objects with users through Social Networks is already implemented. Thus, non-expert users only need to implement the logic needed to manage the Smart Objects' actuators according to a skeleton available in the generated projects.

* Corresponding author.

E-mail address: gonzalezcristian@uniovi.es

In other words, non-expert users do not need to know how to connect Smart Objects to Social Networks, they only need a basic knowledge of programming applications for specific Smart Objects.

This paper address the first part of the research idea presented in [2]. Bilrost not only will address the controlling of actuators through Twitter users' posts, but also will address the communication Object-Object and Object-Human. In the final stages, Bilrost will enable Smart Objects to post messages on Twitter that will invoke the actions of other Smart Objects, and they will be able to share their status with users. Finally, Bilrost will also be able to generate Smart Objects without the necessity of programming skills, achieving the automate completion of the skeletons that are going to be presented in this paper.

In the following lines, the proposal is going to be presented (Section II), addressing what it is, how it works, how its architecture is, how the BSL is. Next, the proposal is going to be evaluated by comparing opinions from two different user profiles after completing an assigned task (Section III). After present and evaluate the proposal, the next section is going to address the related work (Section IV), present the conclusions (Section V), and describe the possible future work that can be done from here (Section VI).

II. Case Study

This section is going to address Bilrost. It was developed to investigate if the communication between objects and people through traditional Social Networks like Twitter is possible.

Bilrost aims to enable non-expert users to generate applications that easily connect people and objects. The goal is to achieve that anybody, without knowledge about how to connect devices to the Internet, will be able to generate applications that connect their devices with them through Social Networks like Twitter. However, to use this proposal, it is required a basic programming knowledge, hence, the goal of Bilrost is to help people with that basic programming knowledge, or non-expert users, to connect their Smart Objects to Social Networks to perform actions according to the messages that their owners sent to the Social Network. For instance, this proposal is suitable for users who have specific knowledge about developing simple applications for a Raspberry Pi, but they do not have enough programming knowledge to develop complex applications that use the Twitter's API.

Twitter was the Social Network chosen due to some interesting features like the lack of reciprocation in the relationships and its mark-up language of tweets (hashtags, mentions, etc.). However, Bilrost is prepared to use more Social Networks in the future.

In short, the main aim of Bilrost is that everybody can handle Smart Objects remotely without specific knowledge. For that, Bilrost generates a skeleton of an application where all the logic needed to connect Smart Objects to Twitter is already implemented, but users must complete it with the logic required for using the actuators of each device.

A. Work-Cycle

Bilrost enables non-expert users to develop applications that handle their Smart Objects' actuators through Twitter. These non-expert users must interact with Bilrost twice to obtain a final application that enables them to handle their Smart Objects through tweets. Thus, the work cycle consists of two steps: **project generation** and **project completion**.

It is important to mention that Bilrost does not generate all logic, but it generates the logic required to connect Smart Objects to Twitter and it creates a skeleton already prepared to be completed with the specific logic of each Smart Object. This skeleton contains empty methods that users must fill as they want with the logic to perform actions. These methods will be called according to the received tweets.

1. Projects Generation

The work cycle starts with the generation of application projects. Firstly, users must write the definition of a device using the BSL syntax (the syntax will be explained in the next section). After writing the definition of the device, the project generator processes the definition to generate an application project where the logic required to establish communication with Twitter is already implemented. Moreover, this application project contains a skeleton that makes easier the specific implementation for each action.

In short, this step consists of two sub-steps:

1. Writing the definition of a Smart Object using the BSL syntax.
2. Using Bilrost to generate the skeleton of the application from the definition written in the previous step.

2. Projects Completion

The work cycle ends with the completion of projects generated in the previous step. The generated projects contain empty methods which users must fill with the logic needed to enable actuators to perform actions.

This step requires basic knowledge about developing applications for Smart Object because users must implement the actions that an actuator can perform. Bilrost does not take part in this step because the proposal is focused on the generation of the logic needed to connect people and Smart Objects through Social Networks. The automation of this step is a part of the future work to take into consideration in future research.

After filling the skeleton, the project is ready to deploy in the target Smart Object.

In short, this step consists of the other two steps:

1. Implement the specific logic to control the Smart Object. Users must implement Smart Object's actions which will be triggered by tweets.
2. Deploy the final application in the Smart Object.

B. Architecture

The architecture of Bilrost can be divided into two components: the **BSL Parser**, and the **Project Generator**.

The first component, the BSL Parser, is responsible for processing the definition of Smart Objects written using the BSL syntax. The result obtained in this component is sent to the second component, the Project Generator, which takes the result processes it to generate a project ready to connect a Smart Object to Twitter. The generated project contains the skeleton that users must fill as it was already said. When users complete the implementation, the program will be finished, and it will be ready to be deployed into devices like Android, Raspberry Pi, or other devices supported by BSL.

1. Bilrost-Specific Language Parser

The BSL Parser is the component responsible for processing definitions written using BSL. It receives a file, written with the BSL syntax, with the definition of a device and generates a tree which contains all required data. This tree will be sent to Project Generator in JSON format.

The content of the file is a model that represents a device with its actuators, the actuators' actions, the Social Networks that it will use with the needed parameters, and more required information that will be explained later.

For instance, Code 1 shows a little example of a device definition using BSL. This definition would be the input of the BSL parser and represents a device whose programming language is Python and has two actuators: a LED and a screen. The LED's actions are '*on*'

and 'off', and the screen's action is 'show'. This device will establish communication with Twitter but only two users will be able to handle it: 'dani_meana' and 'bilrost_bridge'. The rest of the fields are going to be explained in Section II.C.

```
DEVICE IN PYTHON

  FILTER BY 'bilrost', 'uniovi'

  SOCIAL NETWORKS

    CONNECT TO TWITTER

    USERNAME 'username'

    PASSWORD 'password'

    USERS 'dani_meana', 'bilrost_brdige'

  ACTUATORS

  DEFINE 'led'

    LOCATION 'rips'

    ACTIONS 'on', 'off'

  DEFINE 'screen'

    LOCATION 'rpi'

    ACTIONS 'show'
```

Code 1. Definition of a device using Bilrost-Specific Language.

2. Project Generator

The Project Generator is the component responsible for generating projects that already contain the logic needed to establish communications with Social Networks and the skeleton that users must fill.

The Project Generator waits for the tree in JSON format generated in the previous component to generate the application. From the tree, the generator chooses a template that fits the input data and fills it with the data of the tree. The communication with Social Networks and the processing of the data is already implemented so users only must address the concrete implementation of each action.

C. Bilrost-Specific Language

A textual DSL for Bilrost called Bilrost-Specific Language was also designed. Code 2 shows the BSL's context-free grammar written in *Backus-Naur Form* (BNF) although there are tokens that are not explained as WORD or COMMA because they are part of the lexical step. Most tokens were defined to represents the same word as their names except WORD and COMMA. The token WORD represents any string composed of letters, numbers, underscore (_), and dash (-). The token COMMA represents the character comma (,).

BSL does not distinguish between lower case and uppercase, allows writing all code in a single line or multiple lines mixing uppercase and lower case, and changing the order of the different blocks that define a device. Each file written in BSL defines a unique device, hence, users must create as many files as devices they want to define. The definition of a device is composed of the properties of the device and two other different blocks: Social Networks, and Actuators.

To write a program with BSL users must start defining the project language that they want to generate and after that, they must write the properties of the device like filters, Social Networks to connect with their properties, and the actuators.

Furthermore, there are also comments in BSL. The syntax of comments is the same as Python syntax. It starts with a hash sign (#) and ends with a new line.

```
<device>        ::= DEVICE IN <platform> <properties> END
<platform>      ::= PYTHON
                | JAVA
                | ANDROID
<properties>    ::= <property>
                | <properties> <property>
<property>      ::= <filter>
                | <social-networks>
                | <actuators>
<filter>        ::= FILTER BY <filters>
<filters>       ::= WORD
                | WORD COMMA <filters>
<social-networks> ::= SOCIAL NETWORKS <social-networks-list>
<social-networks-list>   ::= <social-network>
                | <social-networks-list> <social-network>
<social-network>  ::= CONNECT TO TWITTER <twitter-properties>
<twitter-properties>     ::= <username> <password> <users>
                | <username> <users> <password>
                | <password> <username> <users>
                | <password> <users> <username>
                | <users> <username> <password>
                | <users> <password> <username>
                | <username> <password>
                | <password> <username>
<username>      ::= USERNAME WORD
<password>      ::= PASSWORD WORD
<users>         ::= ALLOW <users-list>
<users-list>    ::= WORD
                | WORD COMMA <users-list>
<actuators>     ::= ACTUATORS <actuators-list>
<actuators-list>  ::= <actuator>
                | <actuators-list> <actuator>
<actuator>      ::= DEFINE WORD <location> ACTIONS <actions>
                | DEFINE WORD ACTIONS <actions> <location>
<location>      ::= LOCATION WORD
<actions>       ::= WORD
                | <actions> COMMA WORD
```

Code 2. Context-free grammar in BNF.

1. Device Definition

The first step is to define the device's properties. The first property that users must define is the application language. This proposal can generate application projects in Python, Java, and Android.

Another property of a device is the filters. Users must define some keywords that help to identify the device in Social Networks. These keywords can be used to filter the messages that the device search in Social Networks. A device will only perform actions if the messages which will arrive contain the specified filters. A device can have as many filters as users want but at least one.

The next code is the skeleton to define a device.

```
DEVICE IN PYTHON | JAVA | ANDROID

  FILTER BY ...

  SOCIAL NETWORKS ...

  ACTUATORS ...
```

The next code is the skeleton to indicate filters.

```
FILTER BY 'filter1', 'filter2', …
```

After that, users must indicate the Social Networks that they want to connect their device and the actuators that composed the device.

2. Social Networks

This prototype was developed to work with Twitter in the first stage, but it is adaptable to other Social Networks. Hence, BSL allows the definition of several Social Networks. The block to indicate the Social Networks that the device will use starts with the reserved words SOCIAL NETWORKS followed by the parameters required by each Social Network.

The next code is the skeleton to indicate which Social Networks will use the device.

```
SOCIAL NETWORKS
  CONNECT TO TWITTER | OTHERS
```

As mentioned above, this prototype uses Twitter as the Social Network, so the parameters that users must write are their credentials. For that, there are two reserved words USERNAME and PASSWORD. The Project Generator uses these parameters to obtain the tokens required by Twitter API, hence, the final application will not contain the credentials.

Furthermore, there is an optional third parameter to control what users can call the actions of an actuator. It adds a security filter avoiding the control of users' actuators by malicious users. For that, users must use the reserved word ALLOW.

The next code shows the skeleton to configure Twitter as Social Network.

```
CONNECT TO TWITTER
  USERNAME 'username'
  PASSWORD 'password'
  ALLOW 'user1', 'user2', …
```

3. Actuators

Bilrost uses Twitter to invoke the actions of users' actuators. For that, users use BSL to define the devices' actuators with their actions. The block needed to define the actuators starts with the reserved word ACTUATORS and to define each actuator users must write the reserved word DEFINE followed by the name that they want to assign to the actuator. Furthermore, an actuator has several properties that users must define. These properties are the location of the actuator and the name of the actuator's actions.

The location is useful to filter the messages that arrive at the device. In this way, the device could receive messages with a specific location and only the actuators in this location would respond. To specify a location, users must write the reserved word LOCATION.

The name of the actuator's actions is used to enable the invocation of the actuator's actions through Social Networks. Moreover, the name of actions is also the name of the methods that users must fill in the project competition step. A device can have as many actions as users want but at least one.

The next code shows the skeleton to indicate the actuators that compose the device.

```
ACTUATORS
  DEFINE 'name'
    LOCATION 'filter1'
    ACTIONS 'action1', 'action2', …
```

D. Communication Through Twitter

The communication through Twitter is made by tweeting in the timeline. A tweet to control an actuator must contain the device's filters, the actuator's filters, the actuator's name, the action to call, and the parameters that the action could need. Moreover, due to the Twitter limitations (repeated tweets), the messages should contain more content at the final of the message, for instance, the timestamp

To represent the filters of a device, its location, and/or its name, users must use hashtags (#) whereas the action to call must be plain text and its parameters must be enclosed in quotes. Furthermore, users must implement how to parse the parameters in the project completion step.

The hashtag that represents the name is the unique one that is not mandatory. If the name was not specified, all actuators which are in the location would execute the action specified.

In the following lines, there are examples of tweets that handle actuators:

- **#bilrost #uniovi #rpi #red on**: It invokes the action named *on* of an actuator named *red*, located in *rpi*, and it is filtered by the keywords *bilrost* and *uniovi*.
- **#bilrost #uniovi #smartphone #flash on**: It invokes the action named *on* of an actuator named *flash*, located in *smartphone*, and it is filtered by the keywords *bilrost* and *uniovi*.
- **#bilrost #uniovi #rpi off**: It invokes the action named *off* of all actuators located in *rpi*, and they are filtered by the keyword *bilrost* and *uniovi*.

The next examples show the use of actions parameters.

- **#bilrost #uniovi #rpi #display show "hello world"**: It invokes the action named *show* of an actuator named *display*, located in *rpi*, and it is filtered by the keywords *bilrost* and *uniovi*. Moreover, it sends the parameter *hello world* to the action.
- **#bilrost #uniovi #lab #thermostat set "22"**: It invokes the action named *set* of an actuator named *thermostat*, located in *lab*, and they are filtered by the keywords *bilrost* and *uniovi*. Moreover, it sends the parameter *22* to the action.

The filters, the location, and the name of an actuator are the same type of words so the generated application will check all possible combinations that can fit with the defined device.

III. Evaluation and Discussion

This section is going to describe the evaluation process and discuss the obtained results.

A. Methodology

The evaluation process consists of two phases where information was collected to check if the proposal is useful not only for expert users but also for users that have not knowledge about the IoT. In the first phase, it is measured the time that users spent to complete a specific task. After that, in the second phase, users filled a survey based on the Likert scale to measure their opinions about the proposal and its usefulness.

1. Phase 1

In this phase, users of two different profiles had to complete a task that emulated a real scenario that required two actuators connected to Twitter with several actions. The entire task is in the following paragraph.

The required task consisted of defining a device which had two actuators: a fan and a thermostat. The device was a Raspberry Pi and the programming language for the development was Python. The

device would have to look for messages that contained the keywords bilrost, evaluation, and test, the device would have to be connected to Twitter and the only user who would be capable of handling the device would have to be *@bilrost_bridge*. The fan's actions were to turn it on, turn it off, and set its speed. The thermostat's actions were to turn it on, turn it off, increase the temperature, and decrease the temperature. The location of both actuators would have to be the same. When users would have finished the task, they would have to identify the tweets that perform the next actions:

- Turning the fan on.
- Turning the thermostatic on.
- Increasing the temperature.
- Setting the fan speed to 2000 rpm.

Different users were chosen between two profiles, people who had knowledge about the IoT and people without this knowledge because the target of using a DSL is to avoid the requirement of knowledge about a specific technology. A total of 20 participants took part in the evaluation process: 13 participants with knowledge about the IoT and 7 participants without this knowledge but all of them with basic programming knowledge.

During the evaluation, every user had the documentation needed to perform the task where the objectives and the BSL syntax were explained, and some examples were also available to make it easier to understand how the system works. Furthermore, they had time to read the documentation without a limit of time with the possibility of asking any doubt about the system. After that, the system was shown to them and they had more time to test it in order to try to remove learning effects from the gathered results. When users were ready, the task was explained and gave them more time to understand it and think about how to solve them but without access to the system.

Finally, when the participants had said that they are ready and they understood everything, the measurement of times started, and it stopped when users completed the task correctly.

The time limit to complete the task was four times greater than the time spent by the developer of the prototype. It is important to mention that the participants had the documentation available to consult during the evaluation process.

2. Phase 2

After finishing the first phase, users must complete an anonymous survey about this proposal. To create the survey, the 5-points Likert Scale was used because it is the most used in the design of scales. The given options were the following: 1 as strongly disagree, 2 as disagree, 3 as neutral, 4 as agree, and 5 as strongly agree.

The survey was composed of ten declarations that ask users for their opinions about the creation of applications that interconnect objects and humans using this proposal and its possible impact on the IoT.

The survey is composed of a set of ten declarations that are shown in Table I.

TABLE I. Survey Given to the Users

| Declaration	Description
D1	The user understands the functionality of the Domain-Specific Language (DSL) elements and their role in application creation process.
D2	This DSL allows to interconnecting devices and people easily, using a few code lines and spending a little time.
D3	Using a DSL makes it difficult to make mistakes while the user is modelling the applications.
D4	This solution offers a fast way to developing the indicated task.
D5	This solution helps create applications to interconnect objects and people.
D6	The DSL does not require the user to use complex programming skills, as in traditional application development.
D7	The DSL includes enough elements and functionality for the user to create a wide range of applications to interconnect objects and people.
D8	This proposal is a positive contribution to encourage the development of services and applications that provide interconnection between objects and people.
D9	Internet of Things will be benefited by this solution.
D10	This DSL could be used to simplify the classic development process of software applications in other areas.
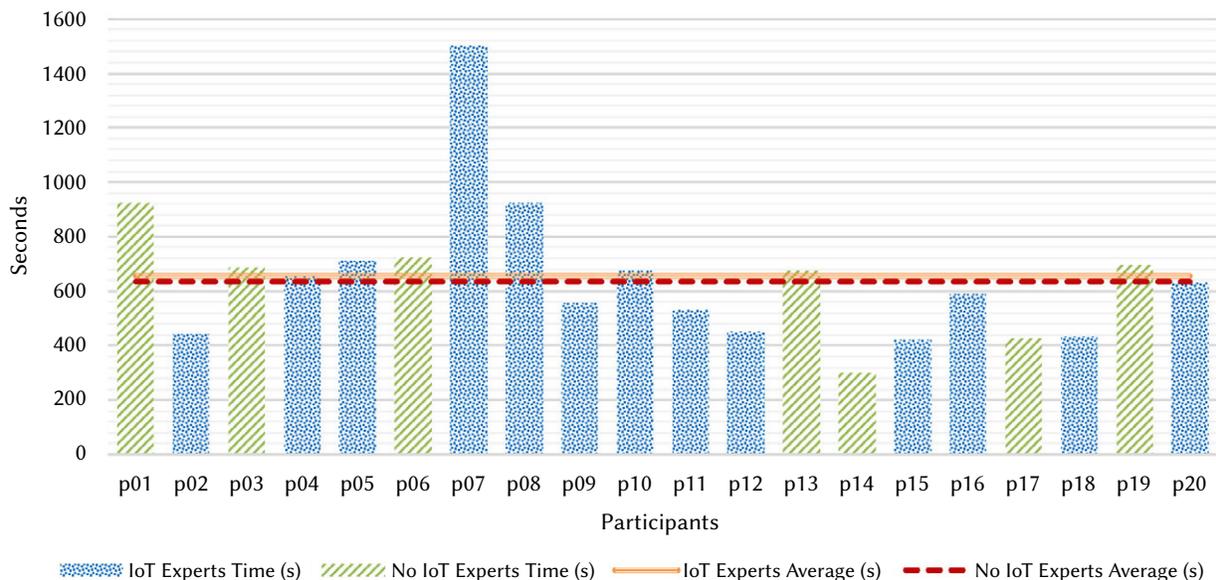
Fig. 1. Time to complete the task per participant.

B. Results

When the evaluation process finished, results to achieve conclusions was obtained. For that, in the following lines, the results obtained through the evaluation process already defined before will be present. The statistical analysis was performed using R version 3.3.2. These results are going to be analysed by an inter-subject study because of the existence of two different groups, users with knowledge about the IoT and users without this knowledge.

1. Phase 1

Table II shows the times obtained in the first phase, the sample size (n), the mean ($\bar{x}$), the standard deviation (s), the maximum and minimum for every profile, and all participants. All-time measures represent seconds spent by users to complete the task.

TABLE II. General Descriptive Statistics of Times Spent By Each Profile

	IoT Experts	No IoT Experts	All participants
n	13	7	20
$\bar{x}$	656.54	634.29	648.80
s	290.80	205.70	258.60
max	1504	926	1504
min	421	303	303

Moreover, Fig. 1 shows the results of the first phase graphically.

2. Phase 2

The second phase or phase 2 consisted of filling a 5-points Likert Scale survey. As has already been explained earlier, the options of the survey were: Strongly Disagree, Disagree, Neutral, Agree, and Strongly Agree. To make it easier to analyse, numeric values were associated with each option from 1 to 5 according to the worst and the best opinion.

Table III shows the responses of each participant anonymously by indicating the profile of each participant, the numeric value of each answer, and the total score of each participant.

TABLE III. Responses of Participants for Each Declaration

IoT Expert	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	Total
p01 No	4	5	5	5	5	4	5	5	5	2	45
p02 Yes	5	5	5	5	5	4	5	5	5	4	48
p03 No	4	5	4	5	5	3	2	5	4	5	42
p04 Yes	4	4	4	5	3	3	5	3	5	4	40
p05 Yes	5	4	5	5	5	2	4	2	2	1	35
p06 No	4	5	4	4	4	3	5	4	3	4	40
p07 Yes	4	4	4	4	4	4	3	5	5	4	41
p08 Yes	4	5	3	4	4	5	4	5	5	4	43
p09 Yes	5	5	4	5	5	5	5	5	4	5	48
p10 Yes	4	5	4	5	5	4	5	5	5	5	47
p11 Yes	5	4	2	5	3	3	4	5	5	2	38
p12 Yes	4	5	2	5	5	5	4	5	5	4	44
p13 No	3	4	4	5	4	5	4	4	4	4	41
p14 No	5	4	3	5	4	5	5	5	4	5	45
p15 Yes	5	4	4	5	4	4	4	5	5	5	45
p16 Yes	5	5	4	5	5	5	5	5	5	5	49
p17 No	4	4	4	4	5	4	4	4	4	4	41
p18 Yes	5	5	4	5	5	5	5	5	5	5	49
p19 No	5	4	5	4	5	4	4	4	4	4	43
p20 Yes	5	5	4	5	4	5	4	5	4	4	45

Fig. 2 shows the distribution of responses for each declaration per user profile. As it shows, most responses are positive although there are some negative opinions.

Table IV shows the descriptive statistics of all declarations and it can be seen the breakdown of each question: the minimum, the first quartile, the median, the third quartile, the maximum, the range (maximum - minimum), the range between quartiles and mode.
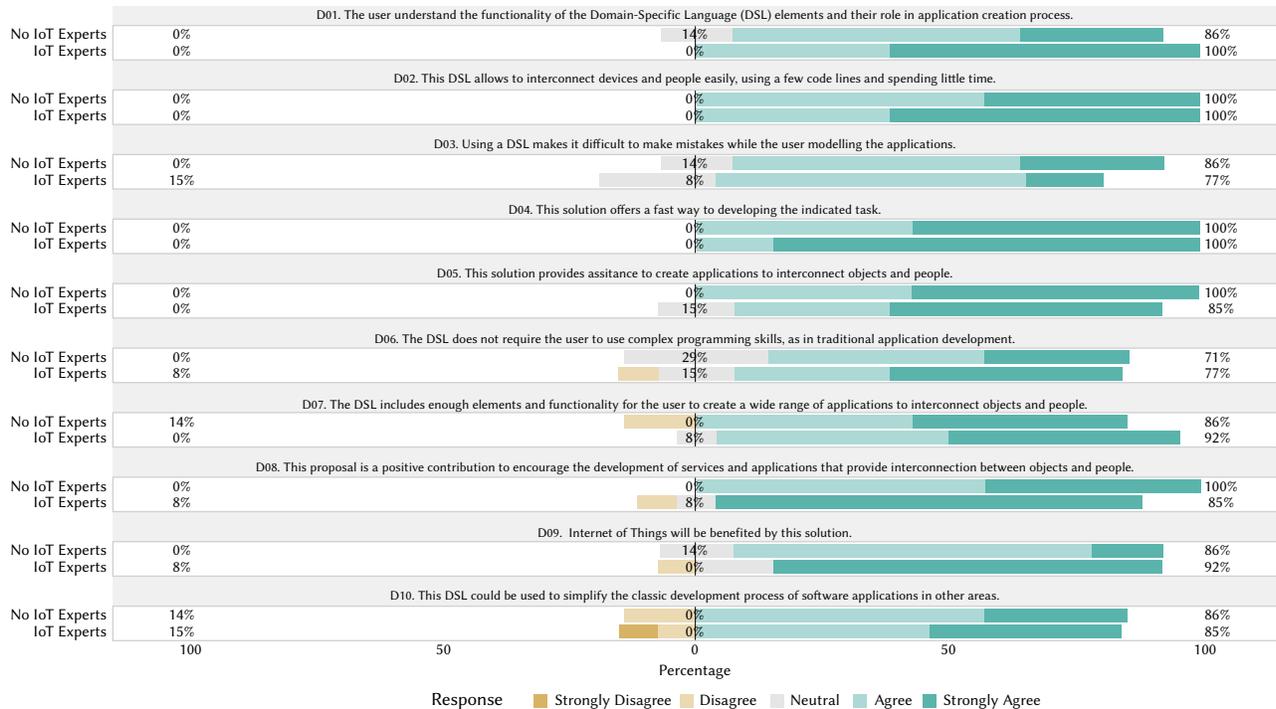

Fig. 2. Distribution of responses for each declaration per user profile.

TABLE IV. General Descriptive Statistics of Each Declaration

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
Min	3	4	2	4	3	2	2	2	2	1
Quartile 1	4	4	4	4.75	4	3.75	4	4	4	4
Median	4.5	5	4	5	5	4	4	5	5	4
Quartile 3	5	5	4	5	5	5	5	5	5	5
Max	5	5	5	5	5	5	5	5	5	5
Range	2	1	3	1	2	3	3	3	3	4
Inter Qrt. - Range	1	1	0	0.25	1	1.25	1	1	1	1
Mode	5	5	4	5	5	5	5	5	5	4

Fig. 3 shows all this data in a Box and Whiskers Plot diagram.
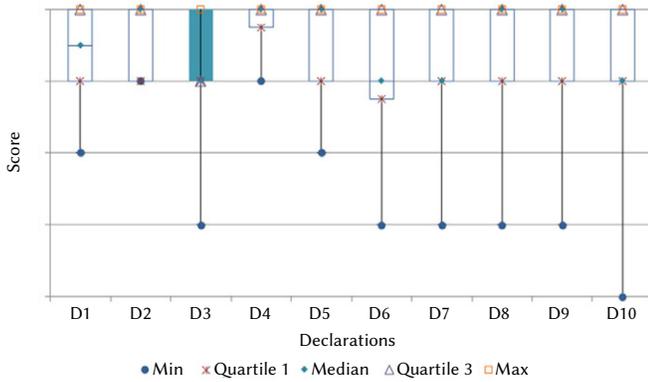
Fig. 3. Box and whiskers plot per declaration.

The frequencies of the responses to each question are shown in Table V. Here, the breakdown of each declaration is shown: the number of votes for each decision and the percentage corresponding to both.

TABLE V. Frequencies Table for the General Responses

		Strongly disagree	Disagree	Neutral	Agree	Strongly agree
D1	#	0	0	1	9	10
	%	0%	0%	5%	45%	50%
D2	#	0	0	0	9	11
	%	0%	0%	0%	45%	55%
D3	#	0	2	2	12	4
	%	0%	10%	10%	60%	20%
D4	#	0	0	0	5	15
	%	0%	0%	0%	25%	75%
D5	#	0	0	2	7	11
	%	0%	0%	10%	35%	55%
D6	#	0	1	4	7	8
	%	0%	5%	20%	35%	40%
D7	#	0	1	1	9	9
	%	0%	5%	5%	45%	45%
D8	#	0	1	1	4	14
	%	0%	5%	5%	20%	70%
D9	#	0	1	1	7	11
	%	0%	5%	5%	35%	55%
D10	#	1	2	0	10	7
	%	5%	10%	0%	50%	35%

Finally, Fig. 4 shows a bar graph with the frequency of the responses in the set formed by both profiles.
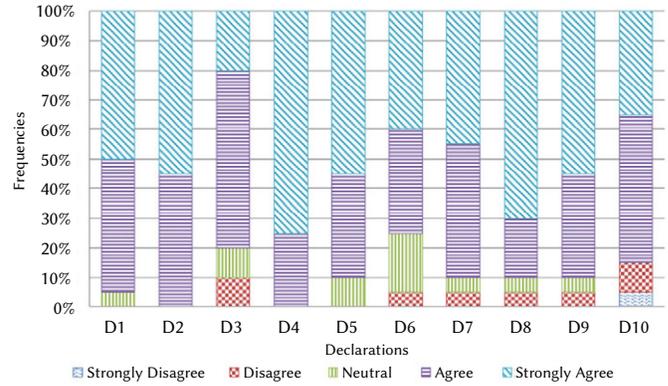
Fig. 4. Frequencies of responses per declaration.

C. Discussion

This subsection includes a discussion to achieve conclusions.

The aim of the data collected from Phase 1 (Table II and Fig. 1) is to conclude if knowing the IoT affects the time spent by users to complete the task. Before performing a statistical test that verifies that hypothesis, it must be determined if the sample data follow a normal distribution and perform a homoscedasticity test.

The application of the Shapiro-Wilk test shows that the data from users with knowledge about the IoT do not follow a normal distribution ($p = 0.001$) whereas the data from the other profile of users follow, indeed a normal distribution ($p = 0.4$).

As one sample does not follow a normal distribution, the next test used was the Levene test to test the homoscedasticity. The result of that test was the homogeneity of variances ($p = 0.7$).

Finally, to conclude if the relationship between having knowledge about the IoT and the time needed to complete the task is significant, the test used was the Mann-Whitney U test which result was that there are no significant differences ($p = 0.5$) between the time spent by users with knowledge about the IoT and the time spent by users without this knowledge.

Fig. 5 shows a non-significant difference in the time spent to complete the task by the two profiles. There are two outliers that represent the fastest participant and the slowest one.
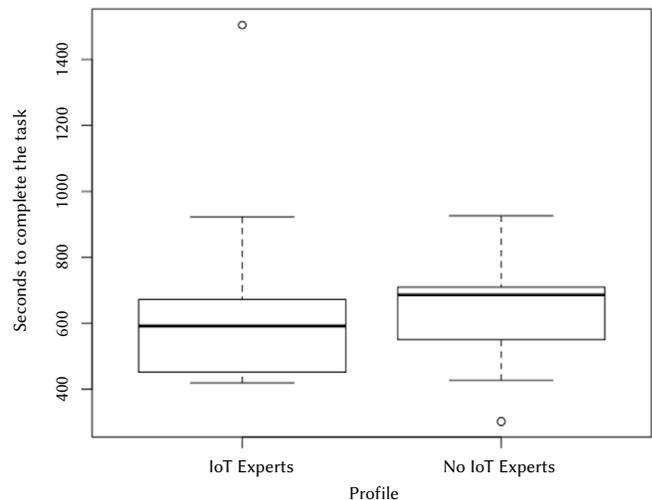

Fig. 5. Box and whiskers plot for times spent by users per profile.

By comparing Fig. 1 and Fig. 5, it can be deduced that the two outliers are the participant p07 and the participant p14. The evaluation process of participant p07 was more laborious than others because he found it hard to understand how the system works. However, participant p14 is a user expert in the development of applications capable of learning a programming language without so much effort. Thus, it was expected that p14 spent less time than other participants to complete the task and it is reasonable to deduce that p07 needed more time than other participants to complete the task.

To conclude the phase 1 of the evaluation process, it can be assumed that having knowledge about the IoT does not affect the use of this proposal because there are no significant differences between both groups. Thus, this proposal is useful for any user without taking into consideration its knowledge about the IoT. However, it is important to remember that this proposal requires basic programming knowledge.

Conclude if the relationship between knowing the IoT and the opinions, expressed via the Likert survey, is significant, is addressed by the collected data in Phase 2.

Nevertheless, it is important to mention that the total score, shown in Table III, is an ordinal variable instead of a cardinal variable because different answers in a Likert Scale do not represent different grades of opinions in an equidistant way, hence, a participant with more score has a better opinion can be assumed but it cannot be said much better is that opinion. Thus, perform statistical analyses that compare the averages to validate the hypothesis is not possible, consequently, a non-parametric test will be used even though the data would fit a normal distribution.

To choose the proper test to perform the statistical analysis, it is necessary to check if the data follow a normal distribution and perform a homoscedasticity test.

The test used to check if the sample data follows a normal distribution was the Shapiro-Wilk test which results was that the data from users with knowledge about the IoT follow a normal distribution ($p = 0.3$) and the data from the other profile of users follow a normal distribution ($p = 0.3$) as well.

As both samples follow a normal distribution, the F test to check the homoscedasticity can be used. The result of this test was the obtained homogeneity of variances ($p = 0.06$).

Finally, to conclude if the relationship between having knowledge about the IoT and the opinions, expressed via the Likert survey, is significant, the Mann-Whitney U test was used because of the homogeneity of variances and the ordinal nature of the values. It was obtained that there are no significant differences ($p = 0.3$) between the opinions of users with knowledge about the IoT and the opinions of users without this knowledge.

Moreover, Fig. 6 also shows that there are no significant differences in the total score obtained by the two user profiles in the Likert Survey.

Now, it can be assumed that having knowledge about the IoT does not influence the users' opinions given via the Likert survey. Thus, the results of the Liker survey can be analysed globally, without discriminating both profiles.

On the other side, from the descriptive statistics of all declaration shown in Table IV and Fig. 3, the following interpretations can be suggested:

- D2 and D4 are the declarations with the highest minimum, in this case, 4 out of 5. This means that all participants agreed with the declaration, at the very least.
- D4 is the declaration with the best score. The first quartile is very close to the maximum value, so the majority chose the maximum option.
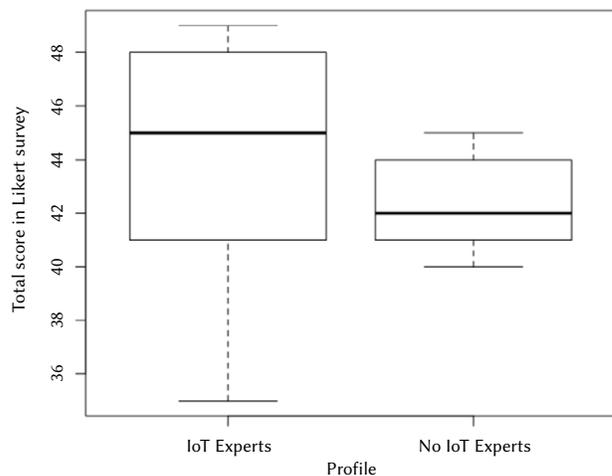
Fig. 6. Box and whiskers plot for total score in Likert survey per profile.

- All questions have a maximum of 5. There is at least one participant that completely agrees with each question.
- D2, D4, D5, D8, and D9 have the highest median, 5 out of 5. From this, it can be deduced that most of the participants agreed with these declarations.
- D2 and D4 have a range of 1 so all participants had the same opinion on these declarations. However, D10 is the only question with a range of 4 which is the worst possible range. It means that there are a lot of differences between the answers of each participant.
- D6 is the question with the biggest dispersion. Only this answer has the first quartile below the answer Agree. Thus, more people chose options different to *Strongly agree* or *Agree* than in the other questions.
- The answer chosen more times is *Strongly agree* because it is the mode of D1, D2, D4, D5, D6, D8 and D9, the mode of D3 and D10 is the answer *Agree* and the mode of D7 are both answers, *Strongly agree* and *Agree*.

Finally, from data shows in Table V and Fig. 4, which contains the frequencies of the responses, it can be figured things that could not be figured before. These are the interpretations:

- D2 and D4 have a 100% of votes for *Agree* and *Strongly agree* and D1 and D5 have 90% or more of votes for *Agree* and *Strongly agree* while only the 10% or less voted *Neutral*. It means that most of the participants agree with theses declarations.
- D7, D8, and D9 have a 10% of votes for *Disagree* and *Neutral*, D3 has 10% of votes for *Disagree* and a 10% of votes for *Neutral*, D6 has less than 10% of votes for *Disagree* but it has a 20% of votes for *Neutral*, and D10 has a 15% of votes for *Disagree* and *Strongly disagree*. It means that the majority agree with theses declarations but there are a few that are indecisive or do not believe in these declarations. Moreover, D10 is the unique declaration with some votes for *Strongly disagree* although the majority votes for *Agree* and *Strongly agree*.

In summary, from the results in Phase 1 we conclude that there are no significant differences between both profiles, and from the results in Phase 2, we conclude a similar result, there are no significant differences between the opinions from both profiles. Users supported this proposal because 80% of the declarations from the survey obtained more than 80% positive or very positive assessments. Moreover, the lowest-rated declaration obtained 75% of positive or very positive assessments.

IV. State of the Art

A. Internet of Things

During the last years, one of the most important topics in research and business is the interconnection between heterogeneous and ubiquitous objects between themselves. This technology is better known as the Internet of Things [3]–[5]. The United States National Intelligence Agency [6] has considered the IoT between one of the six technologies with more interest to the United States from here to 2025.

The aim of the IoT is to interconnect heterogeneous and ubiquitous objects and different systems between each other. To achieve that interconnection, it is required things with Internet capabilities [7]. Thus, it can be considered that the IoT was introduced in order to extend the Internet to things [8]. However, not only the interconnection between objects, but also called Machine-To-Machine (M2M) [9]–[11], is important in the IoT, but also the connections between humans and machines (H2M) [12] and among humans (H2H) [10] are also very important because the three types of communications together allow sharing information between the physical world and the virtual world [13]. This novel proposal tries to offer a way of establishing communications amongst objects, and between objects and humans, by using Social Networks directly instead of a common approach of using web services.

However, there is no single standard or way to do that. There are available different Internet platforms that enable the interconnection amongst objects as can be seen in [3], [14]. Moreover, there are many standards to communicate objects like Near Field Communications (NFC), Radio Frequency Identification (RFID), or Bluetooth. For that, it is necessary to facilitate the development in the IoT.

B. Smart Objects

Smart Objects, also known as Intelligent Products [15], are physical elements that can interact with the environment and/or other objects, have automatic or semi-automatic behaviour depending on the data that they process or receive, and can react according to the interactions with other Smart Objects [4], [16]. Some examples are Smart TVs, smartphones, tablets, some cars, and many other types of devices.

Smart Object can be classified in three dimensions [1] which represent qualities of the object's intelligence: **Level of Intelligence**, **Location of Intelligence**, and **Aggregation level of Intelligence**. The first one indicates how much intelligence an object can have. The second one describes where the intelligence is located. It can be located in the object, in the network, or both. The last dimension indicates if the intelligence is in the element, for example, when the object is composed of various elements and each one has their own intelligence, if it is in the container, or if it is distributed between the container and the elements.

Apart from Smart Objects Not-Smart Objects or objects without intelligence also exist [1]. This type of objects is usually the objects that compose Smart Objects. The Not-Smart Objects are devices that need another device to work like sensors and actuators. Sensors are able to measure physical parameters like the pressure or the temperature, but they cannot process it without another device that is programmed to process data. By the other side, actuators are able to perform actions like control motors or turn on/off lights, but they need another device that orders them the actions to do according to certain conditions.

The proposal of this paper offers Smart Objects whose intelligence is in the container. The other two characteristics depend on the implementation that users developed.

C. Online Social Networks

Online Social Networks (OSN) are valuable resources to develop applications that could be integrated into people's lives. OSNs provide many services that are useful to create applications like identity and authorisation services, APIs to read or write in timelines, receive updates, receive and send private messages and, so on. OSN is a basic piece of Web 2.0 and the convergence of the real world with OSNs enables the development of new applications capable of interconnecting things and humans [17]. Social Networks are commonly used to gather data about people or events for research purposes. For instance, Twitter can be used to extract information about traffic events using Natural Language Processing [18].

Nowadays, there are many OSNs that could be used for researching but research is focused principally on Twitter followed by Facebook. In this proposal, the OSN chosen was Twitter due to its features. Twitter is an OSN and microblogging service based on short messages of up 140 characters very used to research purposes due to several features. Amongst these features are the next: its philosophy of short public messages, it has a specialised markup language that adds semantic information to messages and makes easier the process of the messages, it has a real-time nature, and the relationships no need reciprocation [19]. Users can follow other users without these users follow back. Due to all these features, Twitter is suitable for this proposal, even though it has some limitations.

As stated above, Twitter is very used in research and even in the frame of the IoT. For example, humans can be considered as a type of sensors that can be useful in Smart Cities [20], to detect Earthquakes [21], or also to support smart decisions about the destination of tourism according to the opinions of Twitter's users [22].

The Social Internet of Things (SIoT) is an approach similar to Online Social Networks but focused on objects instead of humans. These social objects are a new generation of objects that can interact with other objects without the intervention of their owners although with their permission. They are capable of discovering other objects, services, and useful information. Moreover, they also can share their services with the rest of the objects in the network [23]. Based on these principles, in [23], they built their own Social Network for Smart Objects. The disadvantages of this SIoT are the dependence on a specific Social Network which is not used for other purpose, and the interaction of this SIoT is amongst objects whereas in this paper is proposed the intercommunication between humans and objects.

Furthermore, scientists of Ericsson [23] observed that people can familiarise better with IoT technologies if there is an analogy between IoT technologies and their habits in OSNs like Facebook, Twitter or any other.

The combination of OSN and SIoT will bring new interesting applications and possibilities for the IoT.

D. Related Work

There are not many similar studies that address the integration of devices on traditional Social Networks. However, some investigations address the communication amongst objects [23], [24] and between people and objects [25].

SenseQ [26], [27] is another approach that interconnects people and objects, and uses Twitter as an interface through which users could make queries using natural language that a Wireless Sensor Network (WSN) would try to resolve by collecting data from interconnected and distributed sensors.

A related work is the Midgar IoT platform [3], [14], [28], [29]. Midgar enables the interconnection between heterogeneous and ubiquitous objects with themselves. It uses a graphic DSL to make easier the creation of this interconnection for people without development knowledge. When users have defined their application, Midgar generates a daemon according to users' definition whose aim

is to monitor the database for changes that indicate how to connect the objects. Moreover, Midgar also includes a graphic DSL to create Smart Objects. However, Midgar needs to use a physical server to work and only enables the communications amongst objects whereas Bilrost does not need any server because of the use of Social Networks to communicate objects and users.

Another platform that enables the generation of interconnected Smart Objects is ELIoT [30]. This research presents a novel programming platform for Internet-connected smart devices that are created using a custom language that is based on Erlang. Like the proposal of this paper, this research provides a new language that makes easier its aim. However, even though this proposal provides more features, the proposal of this paper is focused on the generation of Smart Objects that are connected to Social Networks.

A similar approach is [31]. In this article, authors have created a graphic DSL to allow end-users to define rules for Smart Objects. End-users can define the event or events (using the operators 'and', and 'or') to create the rules composed by the events and its linked actions.

There are many other IoT platforms that allow connecting Smart Objects or things to the Internet. In [32], the authors surveyed about 39 IoT platforms. Nevertheless, they did not mention any platform that allows the generation of Smart Objects that can be handled through Twitter.

An approach similar to this work is Social Access Controller (SAC) [25]. It uses Social Networks to share Smart Objects and enables their management. One advantage of SAC is that it enables not only handling Smart Objects remotely but also sharing their status. However, the use of Social Networks is very different. In [25], Social Networks are used to know the friends of owners of the devices and then, it allows friends to access the Smart Objects through REST architecture. However, the proposal of this paper is different because the access to Smart Objects is through Twitter and users that can handle the Smart Objects do not need to friend anyone, they only need to be mentioned in the application definition written with the DSL.

Another point of view is to use **instant messages** as a way of establishing communication between objects with other objects or humans [33]–[35]. This approach has some disadvantages like the dependency of specific applications which are exclusive for this goal whereas the use of Twitter enables the use of the common Twitter application that it is usually available in so many smartphones and it also enables the use of the web application which prevents from depending on a specific technology.

To summarise, Bilrost is a novel approach that enables communications between humans and objects without many requirements by the humans' side. Humans only need access to Social Networks like Twitter. Moreover, this communication is easy for users that use frequently Social Networks because it is based on the common use of Social Networks, post messages in a timeline.

V. Conclusions

At this stage, the proposal of this paper has already been introduced, Bilrost, a novel one that provides a solution to integrate heterogeneous and ubiquitous Smart Objects into traditional Social Networks. Bilrost enables objects to wait for tweets from users and perform actions according to these tweets.

To achieve an easy integration of objects in Social Networks to people without complex programming knowledge, a new DSL was created, the BSL, which users can use to define their devices with their actuators, the actions that they can perform, and some other properties required to establish the communication with the Social Networks.

Bilrost can generate a project application where the integration into Social Networks is already developed but the specific code for each actuator is not still implemented. Therefore, Bilrost achieves expanding the target audience but it requires that users have a little programming knowledge to fill a skeleton about actuators' actions with a few code lines.

The proposal was evaluated through a two phases evaluation with a sample divided into two different groups with different profiles: IoT experts and no IoT experts. The first phase consisted of performing a task whereas the time spent by the users was being measured and the second one was a Likert survey.

Finally, in the first phase, it was obtained that there are no significant differences between both profiles, and in the second phase it was obtained similar results, there are no significant differences between the opinions from both profiles. Thus, it can be concluded that this proposal is useful for both profiles and their opinions can be analysed altogether. Moreover, in the survey, users supported the proposal because 80% of the declarations obtained more than 80% positive or very positive assessments, and the lowest-rated declarations obtained 75% of positive or very positive assessments. Thus, it can be concluded that Bilrost facilitates the creation of applications that allow non-expert users to control Smart Objects' actuators through Social Networks designed for humans.

Bilrost may be a small step to achieve IoT to be more present in people's day-to-day lives.

VI. Future Work

The Internet of Things is the future so make the integration of IoT technologies in people's diary lives easier is necessary. This proposal follows this way, but it is still not finished, much future work to do from here remains. In the next items, there is some possible future work that arises from this proposal:

- Upgrade the BSL syntax to enable the definition of sensors. This upgrade will increase the possibilities of generated applications.
- Improve the BSL syntax to enable users to define the specific implementation and improve the Applications Generator to generate end-user applications.
- Create a graphic DSL to make easier the generation of applications.
- Compare and study current Social Networks and choose those that are useful to interconnect people and objects.

Analyse different options to secure the com munications through Social Networks because at this stage these communications are public.

References

[1] C. González García, D. Meana-Llorián, B. C. P. G-Bustelo, and J. M. C. Lovelle, "A review about Smart Objects, Sensors, and Actuators," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 4, no. 3, pp. 7–10, 2017, doi: 10.9781/ijimai.2017.431.

[2] D. Meana-Llorián, C. González García, J. Pascual Espada, V. B. Semwal, and M. Khari, "Bilrost: Connecting the Internet of Things through human Social Networks with a Domain-Specific Language," in *Proceedings of the Second International Conference on Research in Intelligent and Computer in Engineering*, 2017, pp. 57–61, doi: 10.15439/2017R110.

[3] C. González García, C. P. García-Bustelo, J. P. Espada, and G. Cueva-Fernandez, "Midgar: Generation of heterogeneous objects interconnecting applications. A Domain Specific Language proposal for Internet of Things scenarios," *Computer Networks*, vol. 64, no. C, pp. 143–158, Feb. 2014, doi: 10.1016/j.comnet.2014.02.010.

[4] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010, doi: 10.1016/j.

comnet.2010.05.010.

[5] K. Gama, L. Touseau, and D. Donsez, "Combining heterogeneous service technologies for building an Internet of Things middleware," *Computer Communications*, vol. 35, no. 4, pp. 405–417, Feb. 2012, doi: 10.1016/j.comcom.2011.11.003.

[6] National Intelligence Council, "Discruptive civil tehnologies - six tehnologies with potential impacts on us interests out to 2025," *Conference Report CR 2008 - 07*. Apr, 2008.

[7] G. M. Lee and J. Y. Kim, "Ubiquitous networking application: Energy saving using smart objects in a home," in *2012 International Conference on ICT Convergence (ICTC)*, Oct. 2012, pp. 299–300, doi: 10.1109/ICTC.2012.6386844.

[8] S. Li, L. Da Xu, and S. Zhao, "The internet of things: a survey," *Information Systems Frontiers*, vol. 17, no. April 2014, pp. 243–259, 2014, doi: 10.1007/s10796-014-9492-7.

[9] E. Borgia, "The Internet of Things vision: Key features, applications and open issues," *Computer Communications*, vol. 54, pp. 1–31, 2014, doi: 10.1016/j.comcom.2014.09.008.

[10] L. Tan, "Future internet: The Internet of Things," in *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, Aug. 2010, pp. V5-376-V5-380, doi: 10.1109/ICACTE.2010.5579543.

[11] M. Hasan, E. Hossain, and D. Niyato, "Random access for machine-to-machine communication in LTE-advanced networks: issues and approaches," *IEEE Communications Magazine*, vol. 51, no. 6, pp. 86–93, Jun. 2013, doi: 10.1109/MCOM.2013.6525600.

[12] International Telecommunication Union, "Overview of the Internet of things," Geneva, p. 14, 2012.

[13] I. Mashal, O. Alsaryrah, T.-Y. Chung, C.-Z. Yang, W.-H. Kuo, and D. P. Agrawal, "Choices for interaction with things on Internet and underlying issues," *Ad Hoc Networks*, vol. 28, pp. 68–90, May 2015, doi: 10.1016/j.adhoc.2014.12.006.

[14] C. González García, J. P. Espada, E. R. N. Valdez, and V. García-Díaz, "Midgar: Domain-Specific Language to Generate Smart Objects for an Internet of Things Platform," in *2014 Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, Jul. 2014, pp. 352–357, doi: 10.1109/IMIS.2014.48.

[15] G. G. Meyer, K. Främling, and J. Holmström, "Intelligent Products: A survey," *Computers in Industry*, vol. 60, no. 3, pp. 137–148, Apr. 2009, doi: 10.1016/j.compind.2008.12.005.

[16] C. Y. Wong, D. McFarlane, A. Ahmad Zaharudin, and V. Agarwal, "The intelligent product driven supply chain," in *IEEE International Conference on Systems, Man and Cybernetics*, 2002, vol. vol.4, p. 6, doi: 10.1109/ICSMC.2002.1173319.

[17] M. Blackstock, R. Lea, and A. Friday, "Uniting online social networks with places and things," in *Proceedings of the Second International Workshop on Web of Things - WoT '11*, 2011, p. 1, doi: 10.1145/1993966.1993974.

[18] P. Anantharam, P. Barnaghi, K. Thirunarayan, and A. Sheth, "Extracting City Traffic Events from Social Streams," *ACM Transactions on Intelligent Systems and Technology*, vol. 6, no. 4, pp. 1–27, Jul. 2015, doi: 10.1145/2717317.

[19] H. Kwak, C. Lee, H. Park, and S. Moon, "What is Twitter, a social network or a news media?," in *Proceedings of the 19th international conference on World wide web - WWW '10*, 2010, p. 591, doi: 10.1145/1772690.1772751.

[20] D. Doran, S. Gokhale, and A. Dagnino, "Human sensing for smart cities," in *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2013, pp. 1323–1330.

[21] T. Sakaki, M. Okazaki, and Y. Matsuo, "Tweet analysis for real-time event detection and earthquake reporting system development," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 4, pp. 919–931, 2013, doi: 10.1109/TKDE.2012.29.

[22] A. Cacho *et al.*, "Social Smart Destination: A Platform to Analyze User Generated Content in Smart Tourism Destinations," in *New Advances in Information Systems and Technologies*, Á. Rocha, M. A. Correia, H. Adeli, P. L. Reis, and M. Mendonça Teixeira, Eds. Cham: Springer International Publishing, 2016, pp. 817–826.

[23] L. Atzori, A. Iera, and G. Morabito, "From 'smart objects' to 'social objects': The next evolutionary step of the internet of things," *IEEE Communications Magazine*, vol. 52, no. 1, pp. 97–105, Jan. 2014, doi: 10.1109/MCOM.2014.6710070.

[24] L. Atzori, A. Iera, and G. Morabito, "SIoT: Giving a Social Structure to the Internet of Things," *IEEE Communications Letters*, vol. 15, no. 11, pp. 1193–1195, Nov. 2011, doi: 10.1109/LCOMM.2011.090911.111340.

[25] D. Guinard, M. Fischer, and V. Trifa, "Sharing using social networks in a composable web of things," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*, 2010, pp. 702–707, [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5470524.

[26] D. Meana-Llorián, C. González García, V. García-Díaz, B. C. P. G-Bustelo, and J. M. C. Lovelle, "SenseQ: Creating relationships between objects to answer questions of humans by using Social Networks," in *Proceedings of the The 3rd Multidisciplinary International Social Networks Conference on SocialInformatics 2016, Data Science 2016 - MISNC, SI, DS 2016*, 2016, pp. 1–5, doi: 10.1145/2955129.2955135.

[27] D. Meana-Llorián, C. González García, B. C. Pelayo G-Bustelo, and N. García-Fernández, "SenseQ: Replying questions of Social Networks users by using a Wireless Sensor Network based on sensor relationships," *Data Science and Pattern Recognition*, vol. 1, no. 1, pp. 1–12, 2017, doi: http://www.ikelab.net/dspr-pdf/vol1-1/dspr-paper1.pdf.

[28] G. Sánchez-Arias, C. González García, and B. C. Pelayo G-Bustelo, "Midgar: Study of communications security among Smart Objects using a platform of heterogeneous devices for the Internet of Things," *Future Generation Computer Systems*, vol. 74, no. September, pp. 444–466, 2017, doi: 10.1016/j.future.2017.01.033.

[29] C. Gonzalez Garcia, L. Zhao, and V. Garcia-Diaz, "A User-Oriented Language for Specifying Interconnections Between Heterogeneous Objects in the Internet of Things," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3806–3819, Apr. 2019, doi: 10.1109/JIOT.2019.2891545.

[30] A. Sivieri, L. Mottola, and G. Cugola, "Building Internet of Things software with ELIoT," *Computer Communications*, vol. 89, pp. 141–153, 2016, doi: http://dx.doi.org/10.1016/j.comcom.2016.02.004.

[31] G. Desolda, C. Ardito, and M. Matera, "Empowering end users to customize their smart environments: Model, composition paradigms, and domain-specific tools," *ACM Transactions on Computer-Human Interaction*, vol. 24, no. 2, 2017, doi: 10.1145/3057859.

[32] J. Mineraud, O. Mazhelis, X. Su, and S. Tarkoma, "A gap analysis of Internet-of-Things platforms," *Computer Communications*, vol. 89, pp. 5–16, 2016.

[33] J. Choi and C.-W. Yoo, "Connect with Things through Instant Messaging," in *The Internet of Things*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 276–288.

[34] S. Aurell, "Remote Controlling Devices Using Instant Messaging: Building an Intelligent Gateway in Erlang/OTP," in *Proceedings of the 2005 ACM SIGPLAN workshop on Erlang*, 2005, pp. 46–51.

[35] A. Roychowdhury and S. Moyer, "Instant messaging and presence for sip enabled networked appliances," *Publisher unknown*, 2001, doi: 10.1.1.116.6212.

Daniel Meana-Llorián

Daniel Meana-Llorián is a Visiting Professor at the School of Computer Engineering of Oviedo, University of Oviedo (Spain). He is a Graduated Engineering in Computer Systems, MSc in Web Engineering, and PhD in Computer Science. His research interests include Mobile technologies, Web Engineering, the Internet of Things, and exploration of emerging technologies related to the previous ones.

Cristian González García

Cristian González García is an Assistant Professor in the Department of Computer Science, University of Oviedo (Spain). He is a Technical Engineer in Computer Systems, MSc in Web Engineering, and PhD in Computers Science. He has been a visiting PhD candidate in the University of Manchester. Besides, he has been in the University of South Florida as visiting professor. He has also been working in different national and regional projects, as well as in projects with private companies. His research interests include the Internet of Things, Web Engineering, Mobile Devices, Artificial Intelligence, Big Data, and Modelling Software with DSL and MDE .

B. Cristina Pelayo G-Bustelo

B. Cristina Pelayo G-Bustelo is a Lecturer in the Computer Science Department of the University of Oviedo. Ph.D. from the University of Oviedo in Computer Engineering. Her research interests include Object-Oriented technology, Web Engineering, eGovernment, Modelling Software with BPM, DSL and MDA.

Juan Manuel Cueva Lovelle

Juan Manuel Cueva Lovelle is a Mining Engineer from Oviedo Mining Engineers Technical School in 1983 (Oviedo University, Spain). Ph. D. from Madrid Polytechnic University, Spain (1990). From 1985 he is Professor at the Languages and Computers Systems Area in Oviedo University (Spain). ACM and IEEE voting member. His research interests include Object-Oriented technology, Language Processors, Human-Computer Interface, Web Engineering, Modeling Software with BPM, DSL and MDA.