


Article

CrowDSL: Platform for Incidents Management in a Smart City Context

Darío Rodríguez-García [†], Vicente García-Díaz [†]  and Cristian González García ^{*,†} 

Department of Computer Science, University of Oviedo, 33007 Oviedo, Spain; uo231097@uniovi.es (D.R.-G.); garciavicente@uniovi.es (V.G.-D.)

* Correspondence: gonzalezcristian@uniovi.es

† These authors contributed equally to this work.

Abstract: The final objective of smart cities is to optimize services and improve the quality of life of their citizens, who can play important roles due to the information they can provide. This information can be used in order to enhance many sectors involved in city activity such as transport, energy or health. Crowd-sourcing initiatives focus their efforts on making cities safer places that are adapted to the population size they host. In this way, citizens are able to report the issues they identify to the relevant body so that they can be fixed and, at the same time, they can provide useful information to other citizens. There are several projects aimed at reporting incidents in a smart city context. In this paper, we propose the use of model-driven engineering by designing a graphical domain-specific language to abstract and improve the incident-reporting process. With the use of a domain-specific language, we can obtain several benefits in our research for users and cities. For instance, we can shorten the time for reporting the events by users and, at the same time, we gain an expressive power compared to other methodologies for incident reporting. In addition, it can be reused and is centered in this specific domain after being studied. Furthermore, we have evaluated the DSL with different users, obtaining a high satisfaction percentage.

Keywords: smart city; internet of things; domain-specific language; model-driven engineering; crowd-sourcing; citizens; incident reporting



Citation: Rodríguez-García, D.; García-Díaz, V.; González García, C. CrowDSL: Platform for Incidents Management in a Smart City Context. *Big Data Cogn. Comput.* **2021**, *5*, 44. <https://doi.org/10.3390/bdcc5030044>

Academic Editor: Min Chen

Received: 31 May 2021

Accepted: 9 September 2021

Published: 16 September 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Over the last few years, the Internet of Things (IoT) has been positioned as one of the disciplines with the greatest impact on the world of communication technologies [1–3], even gaining popularity amongst the public for interconnecting their devices [4], and growing drastically in size and capability [5]. Until recently, the use of the Internet was conceived only as a communication network between humans, so that they could communicate and share information among themselves. This paradigm has progressively evolved towards a new approach in which a notion of interconnected “smart” objects create pervasive computing environments [6], usually to improve people’s lifestyle [7], and other times to improve their work, or even the availability of organizations and their services [8]. For example, the IoT is used to solve problems by automating things or remotely controlling other objects. Following this reasoning, one of the objectives of the IoT is to achieve a state where different objects that share the same environment are able to interact with each other and cooperate with their neighbors in order to achieve common goals [9]. One relevant issue is the nature of the city. The size of the city may be a factor, and the government, the police, or other authoritative bodies may not know where a problem originates, and may lack the information necessary to tackle it. Other times, they may consider that the problem is not important enough in comparison with other problems. Some cities try to improve urban developments or the quality of life of their citizens, and one option is letting people taking part in the processes that address these issues [10]. However, governments need feedback and information from the citizens to do this [11,12]. In other cases, they

could obtain this information using sensors and objects, and automating some processes, which is one of the purposes of the IoT. However, these sensors and objects can be very different [13].

Within the technologies that the IoT encompasses, there are a large number of objects and devices that underline the heterogeneity that exists between the different nodes that we use to communicate. The IoT can be applied to not-smart [13] or inanimate objects such as pallets, boxes containing consumer goods, cars, machines, and fridges, as well as animate “objects” such as animals and humans [14]. To achieve the communication and monitoring of these entities, a wide range of devices are used, such as sensors, actuators or smart tags such as those included in Radio Frequency Identification (RFID) and Near Field Communication (NFC), many of which have integrated intelligence and communication capabilities [15], and other wireless connection technologies embedded in the physical world [16].

We can apply the IoT to different parts of our life. This is why, within the IoT, new concepts have emerged, and have achieved both a strong impact and acceptance by society in developed countries. Terms such as Smart Object, Smart Home, Smart City or Smart Earth are more and more considered due to the benefits they could provide to the future society. All these concepts pursue a common objective: improving the quality of life for users of these technologies. In this paper, we are proposing a solution for smart cities.

In this context, we can find different tools. Many of these tools are focused on optimizing the services that people use, both in a domestic environment and at the level of coexistence in cities. Two clear examples of these are smart homes and smart cities. The term “Smart Home” is used to designate a residence equipped with technology that allows monitoring of its inhabitants and/or encourages independence and the maintenance of good health [17]. Many of these concepts are aligned with other demographic and sociological ones that predict a tendency towards an increasingly aging society, being indispensable to condition the habitable spaces of cities to the social circumstances of each moment. There are many definitions of “Smart City” that have been stated since its inception. These cities are conceived as a space oriented towards the progressive improvement of the government, economy or transport [18]. They also strive to become “smarter”, understanding this smartness as its efficiency, sustainability and capacity [19] and sharing their culture and knowledge by encouraging the citizens to actively participate [20] in the activity of the city. Some studies, such as [21], in which a self-designed framework ranks European cities in order to estimate their degree of environmental sustainability, extract significant findings such as the fact that there is no correlation between a city’s rank score and its population, but wealthier cities perform better in that ranking. These cities will be even more important in the future because large cities are expected to increase their volume of population over the coming decades by a high percentage. According to the forecasts that are collected, in 2050, the number of people living in cities could be placed at 5.2 billion, compared to 2.6 billion that were recorded in 2010 [22]. To summarize, smart cities should combine urban design features and the IoT applications [23] in order to improve their sustainable development.

With this information, it seems obvious that one of the fundamental instruments for the cities of the future is cooperation between citizens in order to achieve an improvement in coexistence and an optimization of services in urban areas, but this is not a simple goal, because in smart city contexts, a variety of players participate with varied interests and expectations [24]. In this paper, we propose the design and development of a platform for reporting and managing the incidents that are originated in the day-to-day of a city. One way of getting that is the use of model-driven engineering (MDE) by implementing a domain-specific language (DSL). This approach can provide multiple advantages to our research. These languages are applied to a concrete domain, abstracting all the information required. There are several projects that use a DSL in order to facilitate the users without development knowledge of certain tasks regarding a specific domain. Furthermore, they can also reduce the difficulty of a task, resulting in a better adaptation to a concrete problem.

Some examples are MOCSL (Midgar Object Creation Specific Language) [25,26], a graphical language designed to create the object creation for use in the IoT, MOISL (Midgar Object Interconnection Specific Language) [27], which allows creating the interconnection amongst smart objects using Midgar, MUCSL (Midgar Use Case Specification Language) [28], with the same purpose but a textual DSL and XPDML (eXtensible Process Definition Markup Language) [29], used in food traceability processes.

Most of the existing projects make use of different approaches and technologies to perform the incident capture, processing and storage process. In order to involve the users in city activity and achieve greater participation, we must make the issue-reporting process as easy and intuitive as possible.

In this work, we have designed a domain-specific language to allow citizens of smart cities to abstract data collection processes and report all the anomalous events originating in their path. Using this DSL, users can categorize the incidents that are reported as well as specify the elements involved in them (means of transport, buildings, human beings, etc.). The goal of this proposal is to make the information accessible to the greatest number of people. Then, our proposal and contribution is the use of Model-Driven Engineering, making use of a **domain-specific language** in order to **abstract and simplify the incidents modeling process**. These languages can provide several benefits to our projects, such as simplicity, usability or integrability [30], because they have notations and constructs tailored toward a particular application domain [31]. Furthermore, DSLs embody domain knowledge, and thus enable the conservation and reuse of this knowledge [32]. Then, using MDE to obtain a DSL, we can facilitate the whole process, from the development until the maintenance. For instance, it allows us to create a DSL with a good abstraction that is very close to the domain. In addition, if in the future it must be changed or extended, an architecture based on MDE allows developers to reuse source code and the different elements (rules, graphics, etc.) using good, simple practices. In addition, it allows the use of new technologies, and easy interactions and interoperability with other applications because we are using different standards that are used in MDE, like eXtensible Markup Language (XML), XML Schema, XML Metadata Interchange (XMI), and Ecore. In the following section, we explain how our platform was designed and the way it works. Furthermore, to know if the contribution is correct and this DSL and the proposal could be applied to improve the smart cities, we have evaluated this DSL, testing it with different users and surveying them with different declarations about the platform, to understand the opinions of the people who could be the users of this proposal.

The rest of the work is structured as follows: Section 2 presents the state of the art of our paper, including the related work. Then, we explain the case of study in Section 3. The following part is focused on evaluating and discussing the results (Section 4) and, finally, we present the conclusions and future work.

2. State of the Art

In this section, we explain some concepts such as smart city, model-driven engineering, domain-specific languages and the related work to present the theoretical frame of our research.

2.1. Smart City

There are many definitions of the smart city concept. In [33], a smart city is defined as the use of smart computing technologies to make the critical infrastructure components and services of a city more intelligent, interconnected, and efficient. Meanwhile, in [34], the authors present this concept as a city performing well in a forward-looking way in six characteristics: smart economy, smart people, smart governance, smart mobility, smart environment and smart living [35], built on the “smart” combination of endowments and activities of self-decisive, independent and aware citizens. From the point of view of information management, smart city is defined in [36] as the use of communication

technologies in order to strengthen the freedom of speech and the accessibility to public information and services.

2.2. Model-Driven Engineering

Model-driven engineering (MDE) proposes the creation of software models as the focus and primary artifacts of development rather than programs [37], with the goal of both simplifying (making easier) and formalizing (standardizing, so that automation is possible) the various activities and tasks that comprise the software life cycle [38]. MDE tools impose domain-specific constraints and perform model checking that can detect and prevent many errors early in the life cycle [39]. This set of technologies is usually the basis for designing and development of the domain-specific languages.

2.3. Domain-Specific Languages

A domain-specific language (DSL) is a language or executable specification language that provides expressive power focused on, and usually restricted to, a particular problem domain [32]. DSLs represent one of the main practical applications of MDE. They provide substantial gains in expressiveness and ease of use compared with general-purpose programming languages in their domain of application [31]. A prerequisite for the design of a DSL is a detailed analysis and structuring of the application domain [40].

2.4. Related Work

Our project focuses on incidents reporting process in a smart city context. However, previously of the development of this project some other platforms had been designed and implanted in several cities. This section shows the main features of some of the most important platforms.

2.4.1. mPASS and WhenMyBus

This project [41] was developed to provide personalized paths to users with special needs. System architecture is composed of two services that work together, mPASS and WhenMyBus. mPASS has the function of collecting data about urban accessibility, both from sensors and data gathered via crowd-sourcing by application users. As a result, the platform obtains georeferenced data, allowing us to identify some urban barriers and providing accessible pedestrian paths to citizens. mPass has an urban accessibility module that helps to define preferences about aPOIs (accessibility Points of Interest). This way, a user can assign states (*neutral, like, dislike, avoid*) to a certain aPOI (*gap, cross, obstruction, parking, surface, pathway, bus stop, bus*) in order to calculate an optimized path for a certain destination. WhenMyBus has been developed to provide useful information to citizens who travel by bus in the city. In addition, WhenMyBus allows users to get real time information about bus availability and equipment, specifying those buses that present some barriers for citizens with disabilities.

2.4.2. CrowdOut

CrowdOut [42] is a platform that allows users to report traffic offenses they witness in real time to map them on a city plan.

The main objective of the application is the reporting of road safety issues (excessive speed, cars illegally parked, signs and signals), grouping them in several categories (road light, pedestrians, parking, road, speed, road flow, cyclist, overtake and others) so that these could be reported to other citizens, representing them in a map and differentiating between dangerous driving areas and safety zones. Application users can select which offenses they are watching at any time. User reports contain a set of attributes that allow the system to locate the place from which the offense has been sent and its description.

2.4.3. FixMyStreet (FMS)

FixMyStreet (FMS) [11,12] is a system that enabled citizens to report, view and discuss local problems such as graffiti, fly tipping, broken paving slabs or street lighting, and to track their resolution by the local government concerned [11]. Problems reported by citizens are delivered to the relevant authority or body via email. Issues are categorized into several categories (*car parking, graffiti, potholes, abandoned vehicles and so on*). If citizens need to report a non-listed issue, they are advised to contact their councils directly [12]. One of the best advantages this project has is the monitoring and evaluation of citizen reports. Thus, the problem originator is contacted by FMS four weeks later to see if the problem has been fixed [11] in order to evaluate the issue state.

2.4.4. CrowdSC

CrowdSC [10] aims to answer questions such as the roads that are damaged in a certain place or the areas that are slovenly in specific locations. In this way, a council asks for citizens collaboration in order to know the state of a certain point of a city. Authors divide the whole process into three other sub-processes: data collection, data selection and data assessment. The data collection phase consists in querying just a few citizens, who provide a set of images of damaged roads or similar. In this phase only some citizens are queried to avoid receiving subjective or mistaken judgments from them. In second phase, known as data selection, some other citizens are asked to select the most representative photo for each location. In the last phase, data assessment, other citizens assess the priority of each selected photo.

2.4.5. Comparison

All studied platforms have similar features and functionalities:

- They all focus on incidents management in a city context using several technologies and approaches to achieve their objectives. For example, the use of maps is common in all solutions in order to represent the incidents or events reported by users.
- Another common feature in many projects is the fact that many of them incorporate a user profile to track the user activity in the platform.
- Except in one article, they did not test the application with the end-user, who is the most important part. These users have to use the application to send the problem in the cities. The other one did, but they did not describe everything because of confidentiality, and the rest are opinions about the application.

Next, the differences with our proposal:

- It includes many of these features.
- The main contribution of our paper is the use of MDE to create a DSL. With these two concepts, we can obtain a system that can reuse the previous knowledge, abstracting the problem and focusing on the simplicity and usability as stated by [30,32].
- We have evaluated the platform with users. Then, we have obtained results to know the opinion of the people who would have to use the DSL, which has been developed for them. Furthermore, we have used an evaluation that was used in other articles.

Note that not using MDE or a DSL are not intrinsic limitations, but they are usually the preferred options in the software engineering community to create an adequate 'language' to allow users to work in a specific domain.

In addition, with the use of MDE, it is possible to solve or minimize some of the main problems of software development (low quality software, incorrect estimations, maintenance, interoperability, etc.). Then, this is a way of how to design and create software. Besides, applying MDE, we can do a very deep study of the necessities for solving a problem and the stages that can be automated. After that, it is possible to create a very good DSL because you have all the information of the problem.

With the use of models, we can create a tailored language which can allow the reusability in the source code, can automate different phases, can abstract more the problem and

different elements, and can be very oriented to the users, facilitating the use of that application.

Then, the created DSL represents the problem that was studied with MDE and allow users to use that exact part. The problem here is whether the DSL adequately represents the specific domain.

Then, how can we measure the quality of this research, and how can we know if the DSL has been created correctly and represent the intended problem? By using the data of the 'editor' to know if this work is correct according to the use of it and the opinion of the participants.

According to the related work, they did not test the systems with users, so they may create an application that users could like or dislike. In our case, we have created a DSL to solve the problem of reporting incidents in smart cities and we have tested with users as to whether this is a good way or not to solve this problem.

3. Case of Study: CrowDSL

The aim of our proposal is to allow Smart City citizens to model all the entities, scenarios and sectors of the city that are involved in the occurrence of an incident. In order to carry out the report of the aforementioned incidents, we have developed a web editor that allows users to quickly become familiar with the environment and gradually acquire the knowledge necessary to operate in a normal way with the domain-specific language that was developed. Once the incidences are reported and validated, they are published in the same platform.

3.1. Architecture

The platform of our research is structured in three layers that cover the full functionality of the project. Figure 1 shows a diagram of our proposed architecture. The first layer, "Data definition", deals with the models definition process. This layer is composed of the DSL that can be used by means of the editor. In this first phase, we obtain an XML document that contains the entities data and the relationships among them modeled by the user. This information is sent to the second layer, "Data Processing", where the XML document is used to represent a tree in memory that replicates the structure of the file received from the upper layer. This second layer has the objective of validating that model structure complies with the specifications defined in the application meta-model in multiple aspects, such as the entities relationships and data types specified by application users. Once the entire validation process has been completed, the incidents are stored in JavaScript Object Notation (JSON) format in the application database. When the incidents are stored, they are published through the third layer of the architecture, "Data visualisation", which allows the rest of the users of the application to consult all the attached information. Multiple forms of information are proposed for this third layer: the information specified by the user linked to each entity in plain text, multimedia elements provided by the user who reports the incident (images, videos, tweets or geolocation), a parameterized map that allows users to know the location of the incident and, finally, a comments section for each user to give a personal view.

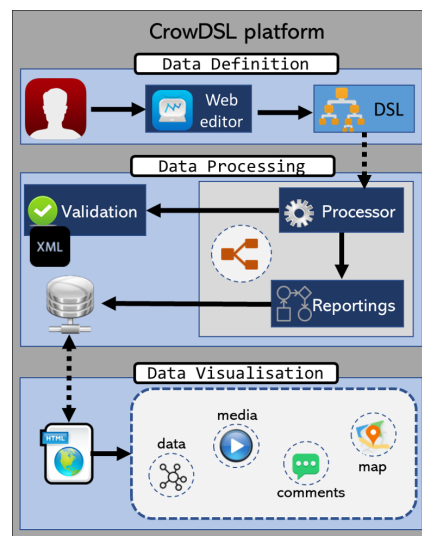


Figure 1. CrowDSL architecture.

3.2. CrowDSL: The Domain-Specific Language

The first step was the definition of the application meta-model. This meta-model contains a wide range of elements that try to represent all existing entities, sectors and scenarios of a Smart City. The meta-model consists of 5 sets of elements, including general elements (user, city, map), alert types (flood, fire, accident, damages, ...), objects (person, animal, house, car, ...), city sectors [35] (economy, people, governance, mobility, environment, living) and multimedia elements (images, videos, geolocation, tweets, ...).

Following one of the principles of MDE, the next step was to define the concrete syntax of elements from the abstract syntax. Table 1 shows the correspondence between the two syntax for some of the main elements of language.

Table 1. CrowDSL abstract and concrete syntax.

Abstract Syntax	Concrete Syntax
User	
City	
Map	
Fire (Alert)	
Person (Object)	
Mobility (Sector)	
Image (Media)	

As has been mentioned previously, we have developed a web editor to work with CrowDSL. Figure 2 shows the look and feel of the web editor. This editor is composed of

multiple sections that allow users to specify all the information related to the incidence they are reporting at any time. Now, we detail the purpose of each section:

- A—Palette of components. With this tool, users drag and drop components to the editor and define relationships among elements.
- B—Editor. This element allows users to watch the incident structure they are reporting at that time.
- C—Properties. This panel shows all properties of the component selected in the editor. This way, users can provide values to entities attributes.
- D—Log. This element shows the feedback to users. Once incidents are sent, they can contain validation errors. These errors are displayed through this component.
- E—Incident representation. Once an incident is validated, a JSON format representation of it is shown in this box to check all data and structure specified.

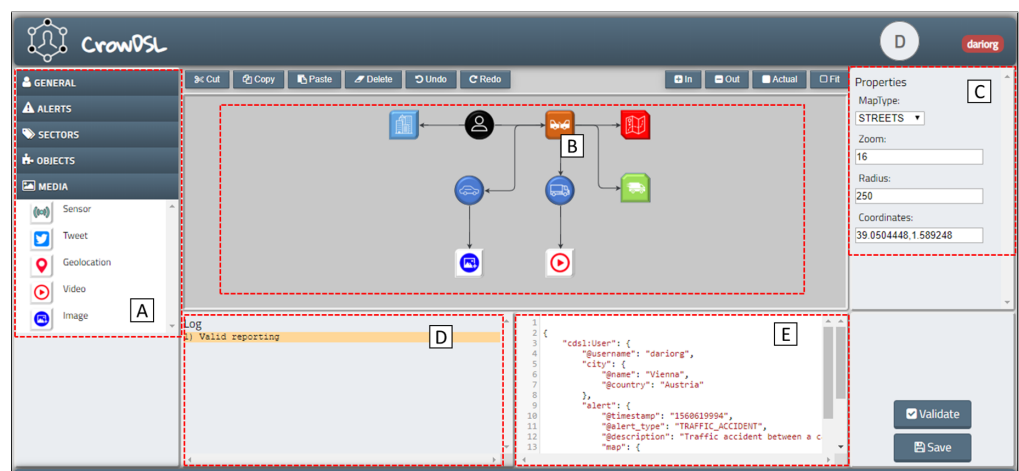


Figure 2. CrowDSL Web editor.

Now, we are going to explain the incidents validation process since they are modeled in the editor until they are stored in a database and published for other users.

3.3. Incidents Validation Process

In Figure 2, where the web editor is displayed, an example of an incident is presented. This incident describes a traffic accident where a car and a truck are involved. The incident representation contains several elements: the user of the application who reports the incident (black circle), the city where the incident occurs (blue square), the type of incident (traffic accident, orange square), the map on which the incident will be represented (red square), the “objects” implied (car and truck, blue circles), the sector of the city affected and the media elements provided by the user (image and video, white squares). Figure 3 focuses on the graphical representation of this incident.

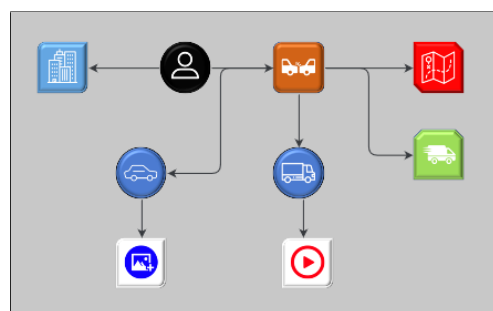


Figure 3. Example of incident reporting.

As users model the relationships between the different nodes that make up the language, an internal representation of the model is generated in an intermediate XML format. When the user clicks the “Validate” button, the process of validation of structure and data of the model is triggered. Validation process is done by means of an XML Schema for an XMI document that represents the meta-model of the application. This XML Schema follows the XMI standard [43] and allows us to perform all validations specified during the meta-model design phase. Therefore, the first step is to make a transformation between the native XML to an XMI format to validate each model against the XML Schema for the XMI obtained. This conversion is materialized by first representing the source XML as a tree in memory in which each node is generated by means of a model class that defines the structure of the element. Starting from that representation, an XMI file is generated by using a visitor design pattern over the tree in order to perform the aforementioned validation. If, during the process, an error is detected in the validation process, the error is represented in the editor to provide a feedback to the user. If the validation process concludes with no errors, the user is allowed to send the incident using the “Save” button. When the structure and content of an incident is validated, a representation of it is generated in JavaScript Object Notation for Linked Data (JSON-LD) format (Listing 1) that facilitates its storage in a NoSQL documents store database to be used later. Figure 4 shows the resulting graphical representation of the incident. This information will be published to all platform users.

Listing 1. Incident representation in JSON-LD format.

```
1  {
2    "cdsl:User": {
3      "@username": "dariorg",
4      "city": {
5        "@name": "Vienna",
6        "@country": "Austria"
7      },
8      "alert": {
9        "@timestamp": "1558255082",
10       "@alert_type": "TRAFFIC_ACCIDENT",
11       "@description": "Collision between a car and a truck",
12       "map": {
13         "@map_type": "STREETS",
14         "@zoom": "15",
15         "@radius": "250",
16         "@coordinates": "48.210033,16.363449"
17       },
18       "service": {
19         "@service_type": "MOBILITY"
20       },
21       "object": [
22         {
23           "@name": "BMW",
24           "@description": "Serie 3",
25           "@danger_level": "NO_DANGER",
26           "@object_type": "CAR",
27           "data_object": {
28             "@url": "https://db.com/image"
29           }
30         },
31         {
32           "@name": "Mercedes",
33           "@description": "Benz",
34           "@danger_level": "NO_DANGER",
35           "@object_type": "TRUCK",
36           "data_object": {
37             "@url": "https://db.com/video"
38           }
39         }
40       ]
41     }
42   }
43 }
```

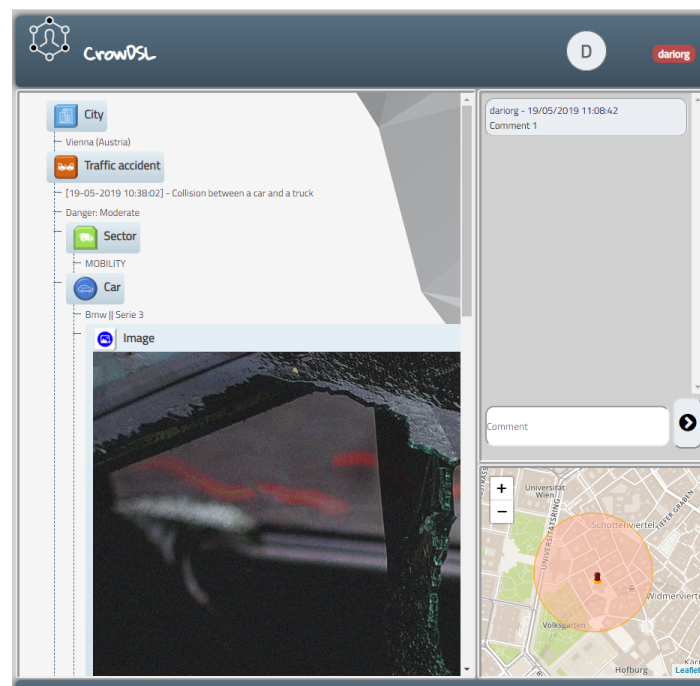


Figure 4. Visualization of an incident.

4. Evaluation and Discussion

In this section, we explain the process followed to evaluate our proposal by introducing the methodology used and the results obtained.

The objective of this proposal was to verify if it is possible to abstract the incidents capture process in a smart city context based on the MDE approach by means of a domain-specific language.

4.1. Methodology

In order to validate our research questions, we used a methodology focused on the evaluation of DSLs. This methodology has been used in other works such as [27,28]. In this way, we asked a set of 20 users for evaluating our prototype following a practical example. They are between 20 and 40 years old. Users who participate in evaluation have different levels of knowledge in the field of IT (Information Technology) and the Internet of Things, representing a wide range of profiles: beginner, middle-level and professional users. Table 2 shows the detailed proportion of users according to this. It is very important to have users of different levels in the IT and in the IoT because they could be the people who manage this type of systems, they are the target population [44]. Then, in this case, it is very interesting to study people with different levels of knowledge in IT and the IoT to know if they can use it and whether they understand everything.

Table 2. Level of knowledge of the users in IT and the IoT.

Level/Type	Beginner	Middle-Level	Professional
IT	4	7	9
IoT	13	5	2

The practical example was as follows:

“User witnesses a traffic accident involving two vehicles, a car and a truck. The collision takes place in the city of Vienna (Austria). The severity of the accident can be considered as high. The car suffers serious damages, and the truck is damaged moderately. The incident radius of the accident is 50 m”.

The focus of the evaluation is to validate the concrete syntax of the DSL, and which will also indirectly validate its abstract syntax and therefore its semantics, which together make up the main parts of the DSL from the point of view of the MDE approach. Before the beginning of the evaluation, each user was instructed in making use of both the DSL and the editor. Explanation just took a few minutes and contained a tiny set of operations with the editor, such as add and connect elements, specify attributes of objects and interpret log errors, based on another small and similar practical case. To allow users to include multimedia elements, an image and a video were provided to them. These elements were supposed to be captured by users in the place of the incident.

Once users finished the incident modeling, they were asked for filling out a survey to measure their degree of satisfaction with the platform. Thus, in order to establish a minimum and a maximum value for the responses it has been used the Likert Scale [45], giving the users the following options to evaluate each declaration: 1 as strongly disagree, 2 as disagree, 3 as somewhat agree, 4 as agree, and 5 as strong agree. Table 3 shows all the responses provided by users for each declaration (the survey contains the declarations showed in Table 4).

Table 3. Users responses to survey declarations.

D/U	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
U1	4	3	5	3	5	5	5	2	5	4
U2	5	4	5	4	5	5	4	3	5	5
U3	5	4	5	3	5	5	5	1	5	4
U4	5	4	5	3	5	5	5	2	5	4
U5	5	3	5	4	4	5	5	2	4	4
U6	4	4	5	3	5	3	4	1	4	4
U7	5	5	5	3	5	4	4	0	5	5
U8	3	5	5	3	5	5	4	2	5	3
U9	5	5	4	5	5	5	5	3	4	4
U10	5	4	5	3	5	5	5	1	4	5
U11	5	5	4	3	5	5	5	0	4	5
U12	4	4	5	4	5	5	5	3	5	5
U13	3	3	5	4	5	4	4	4	5	4
U14	5	3	5	3	5	4	4	5	5	4
U15	4	4	5	3	4	4	4	1	5	5
U16	4	5	5	4	4	4	3	2	5	4
U17	5	4	5	5	5	5	4	2	5	4
U18	5	4	5	4	5	5	4	2	5	5
U19	5	5	5	4	5	5	5	3	5	5
U20	5	4	5	4	5	5	5	1	5	5

4.2. Results

Once we gathered all data, we studied several descriptive statistics applied for all the set. We analyzed the minimum, quartile 1, median, quartile 3, maximum, range, interquartile range and mode of each declaration of the survey. Then, we represent all data using a box and whiskers plot diagram. Table 5 and Figure 5 show the results.

Table 4. Survey declarations.

#	Declaration
D1	The elements of the language are well defined and their purpose are understood by users.
D2	The elements of the language are adequate to model any type of incidence.
D3	It is easy to specify the values for the attributes of elements (level of danger, description, ...).
D4	The logical structure of the relationships between entities is understood.
D5	You prefer this methodology rather than sending the incidents in text mode
D6	The editor elements are well structured and it is easy to work with it.
D7	The web editor is useful for modelling city incidents.
D8	The user understands the log errors of the application.
D9	This work would be useful in a Smart City.
D10	The degree of satisfaction with platform is high.

Table 5. Descriptive statistics.

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
Min	3	3	4	3	4	3	3	0	4	3
Quartile 1	4	4	5	3	5	4	4	1	4.25	4
Median	5	4	5	3.5	5	5	4.5	2	5	4
Quartile 3	5	5	5	4	5	5	5	3	5	5
Max	5	5	5	5	5	5	5	5	5	5
Range	2	2	1	2	1	2	2	5	1	2
Inter Qrt.-Range	1	1	0	1	0	1	1	2	0.75	1
Mode	5	4	5	3	5	5	5	2	5	4

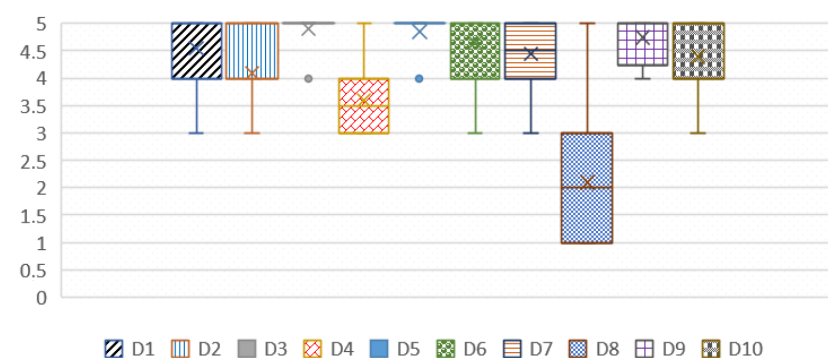


Figure 5. Descriptive statistics in box and whiskers plot diagram.

By analyzing the results, we can do some interpretations:

- D3, D5 and D9 have the highest minimum, so the vast majority of people agreed with those ones, at least.
- D1, D3, D5, D6 and D9 have the highest median, so we can deduce that the users agreed with these declarations.

- D3, D5 and D9 have a range of 1. The meaning of this is that the participants have almost the same opinion in these declarations. On the other hand, D8 has a range of 5, the highest. This means that in D8, there exists a big difference between the answers of the participants. Here, we can find answers from strongly disagree to strongly agree.
- All declarations have a high maximum, with a value of 5 (strong agreement).
- The mode is between 4 (agree) and 5 (strongly agree) in 8 of the 10 declarations, so mostly, users agreed or strongly agreed with those statements. On the other hand, we can find 2 declarations with a mode of 2 in D8 and 3 in D4. Then, the most chosen answer in D8 was *disagree* and in D4 was *neutral*.
- The negative side of this study is shown in D8, where users estimate that the log errors of the application are not understandable enough. This is why the mode and median have a value of 2 (disagreement). Besides, according to D4, the logical structure could be improved because users are between *neutral* and *agree*.

Regarding the frequency of answers by declaration, we tried to find some other patterns that could not be interpreted in the data presented until now. Table 6 and Figure 6 shows the frequency of answers by each declaration in survey.

Table 6. Frequency of answers by declaration.

Declaration		Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
D1	#	0	0	2	5	13
	%	0%	0%	10%	25%	65%
D2	#	0	0	4	10	6
	%	0%	0%	20%	50%	30%
D3	#	0	0	0	2	18
	%	0%	0%	0%	10%	90%
D4	#	0	0	10	8	2
	%	0%	0%	50%	40%	10%
D5	#	0	0	0	3	17
	%	0%	0%	0%	15%	85%
D6	#	0	0	1	5	14
	%	0%	0%	5%	25%	70%
D7	#	0	0	1	9	10
	%	0%	0%	5%	45%	50%
D8	#	7	7	4	1	1
	%	35%	35%	20%	5%	5%
D9	#	0	0	0	5	15
	%	0%	0%	0%	25%	75%
D10	#	0	0	1	10	9
	%	0%	0%	5%	50%	45%

In this way, we can get new interpretations of data:

- Users strongly agree in a very high percentage in D3 (90%), highlighting the facility to provide data to language attributes of elements. Then, a total of 18 users out of 20 strongly agreed with D3, and 2 out of 20 agreed.
- In D5, 85% of users (14 out of 20) prefer this reporting methodology rather than sending incidents in text mode. In total, 4 users agreed and 1 chose neutral.
- D9 has very good acceptance because 75% of the users *strongly agree* with it and the other 25% agree.
- Half of all users (50%) agreed in being satisfied with the platform, and 45% strongly agree with this statement.
- D8 is the worst rated because 70% of the users *strongly disagree*.

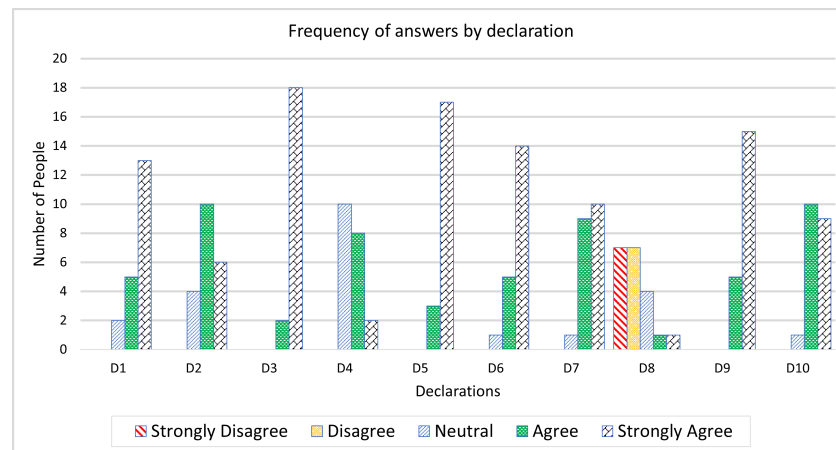


Figure 6. Frequency of answers by declaration.

5. Conclusions

This research aimed to provide a new method for incidents reporting to citizens of smart cities. Then, in this proposal, the main contribution, compared to the other studied platforms, is the use of model-driven engineering by designing and building a domain-specific language in order to tag, link and process all information gathered via crowd-sourcing by users. This approach allowed us to provide many features to our application such as reusability, interoperability and portability. In this way, information could be used by several technologies because information is stored using JSON format, ensuring its interoperability, as well as the use of other standards like XML, XMI, and more. Once we finished the platform development and its future assessment, and to validate the technical effectiveness of this DSL, we performed an evaluation with users. This is the other contribution of this paper. In our case, we have evaluated this DSL testing it with different users and surveying them with different declarations about the platform to know the opinion of the people who could be the users of this proposal. In this survey, we checked that the vast majority of users showed a very good opinion about our DSL, with 95% of satisfaction with the editor (D7). Besides, from a general view, 81.5% of the answers are *Agree* and *Strongly Agree*, with 29% and 52.5% respectively, and 7% are *Disagree* and *Strongly Disagree*, 3.5% each one. Users highlight in the survey some features and characteristics such as the usability, utility and well design of the solution, which are represented in the declarations D1, D3, D5, D6 and D9. Furthermore, the vast majority of people believe that the DSL is easy to specify the values for the attributes of elements, they prefer to use the DSL instead of the text mode, and they believe that this work could be useful for smart cities, according to the results of D3, D5 and D9. Then, according to these results, people with a beginner (4) and middle-level (7) knowledge of IT and with a beginner (13) knowledge about the IoT have been able to use this DSL and the platform.

In summary, we suggest that the use of MDE to create a DSL is a good idea to solve the problem of the communication of incidents in a smart city. Until the moment of this article, some applications appeared, but they did not test with real users, and they did not apply some specific strategy to create them.

However, there are also some features of the DSL and the platform that could be improved in the future. For instance, the log system, the possibility of adding new elements to the language in the future, and the logical structure of the relationships between the entities because these have been the worst rated declarations (D4, D8). In addition, CrowdDSL relies on standard technologies like Python, XML or JSON and well-established design patterns and frameworks. So, the data scale it can handle depends mainly on the specific features of each computer that uses it.

6. Future work

Notwithstanding, this work can be improved and expanded. Then, some future works are:

- Gather data to make statistics analysis about the location of the incidents.
- Statistics analysis of the application logs to improve user interactions and the usability. Besides, it could do a usability study about the current interface.
- Add new elements to the DSL: some users expressed that they believe that other elements could be useful for some cases. For instance, one of them could be the element 'Others' to try to represent different entities that currently have no representation.
- Extend the DSL: currently, the incidents are internally represented using JSON-LD. An option could be to add functionalities to allow searching and filtering of the stored data.
- Data scale tests: carry out load tests and other types of tests to obtain concrete values about the data scale the language can handle.

Author Contributions: Conceptualization, C.G.G., V.G.-D. and D.R.-G.; methodology, C.G.G.; software, D.R.-G.; validation, C.G.G. and V.G.-D.; formal analysis, C.G.G.; investigation, D.R.-G.; resources, D.R.-G.; data curation, D.R.-G.; writing—original draft preparation, D.R.-G.; writing—review and editing, C.G.G. and V.G.-D.; visualization, D.R.-G.; supervision, C.G.G. and V.G.-D.; project administration, C.G.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DSL	Domain-Specific Language
FMS	FixMyStreet
IT	Information Technology
IoT	Internet of Things
MDE	Model-Driven Engineering
MOCSL	Midgar Object Creation Specific Language
MOISL	Midgar Object Interconnection Specific Language
MUCSL	Midgar Use Case Specification Language
NFC	Near Field Communication
POI	Point of Interest
RFID	Radio Frequency Identification
XMI	XML Metadata Interchange
XML	eXtensible Markup Language
XPDMML	eXtensible Process Definition Markup Language

References

1. Council, N. Disruptive Civil Technologies: Six Technologies with Potential Impacts on Us Interests Out to 2025. 2008. Available online: <https://irp.fas.org/nic/disruptive.pdf> (accessed on 16 September 2021).
2. Property, I.; Informatics, O. Eight Great Technologies The Internet of Things. Available online: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/343879/informatics-internet.pdf (accessed on 16 September 2021).
3. Walport, M. *The Internet of Things: Making the Most of the Second Digital Revolution*; UK Government Chief Scientific Adviser. 2014. Available online: <https://www.oxfordmartin.ox.ac.uk/publications/the-internet-of-things-making-the-most-of-the-second-digital-revolution/> (accessed on 16 September 2021).
4. Meana-Llorián, D.; García, C.G.; Pelayo G-Bustelo, C.; Cueva-Lovelle, J.M. BILROST: Handling Actuators of the Internet of Things through Tweets on Twitter using a Domain-Specific Language. *Int. J. Interact. Multimed. Artif. Intell.* **2021**, *6*. [CrossRef]

5. Yu, R.; Kilari, V.T.; Xue, G.; Yang, D. Load Balancing for Interdependent IoT Microservices. In Proceedings of the IEEE INFOCOM 2019—IEEE Conference on Computer Communications, Paris, France, 29 April–2 May 2019; pp. 298–306. [CrossRef]
6. Miorandi, D.; Sicari, S.; Pellegrini, F.D.; Chlamtac, I. Ad Hoc Networks Internet of things: Vision, applications and research challenges. *Ad Hoc Netw.* **2012**, *10*, 1497–1516. [CrossRef]
7. Jin, W.; Xu, R.; You, T.; Hong, Y.G.; Kim, D. Secure Edge Computing Management Based on Independent Microservices Providers for Gateway-Centric IoT Networks. *IEEE Access* **2020**, *8*, 187975–187990. [CrossRef]
8. Cabrera, E.; Cárdenas, P.; Cedillo, P.; Pesántez-Cabrera, P. Towards a Methodology for creating Internet of Things (IoT) Applications based on Microservices. In Proceedings of the 2020 IEEE International Conference on Services Computing (SCC), Honolulu, HI, USA, 18–20 September 2020; pp. 472–474. [CrossRef]
9. Atzori, L.; Iera, A.; Morabito, G. The Internet of Things: A survey. *Comput. Netw.* **2010**, *54*, 2787–2805. [CrossRef]
10. Benouaret, K.; Valliyur-Ramalingam, R.; Charoy, F.; Karim Benouaret, R.V.R.F.C. CrowdSC: Building Smart Cities with Large Scale Citizen Participation. *IEEE Internet Comput.* **2013**, *17*, 57–63. [CrossRef]
11. King, S.F.; Brown, P. Fix My Street or Else: Using the Internet to Voice Local Public Service Concerns. *Comput. Soc.* **2007**, *72*–80. [CrossRef]
12. Baykurt, B. *Redefining Citizenship and Civic Engagement: Political Values Embodied in FixMyStreet.com*; Anais do AoIR—Association of Internet Research: Seattle, WA, USA, 2011; pp. 1–18.
13. González García, C.; Meana Llorián, D.; Pelayo García-Bustelo, B.C.; Cueva Lovelle, J.M. A review about smart objects, sensors, and actuators. *Int. J. Interact. Multimed. Artif. Intell.* **2017**, *4*, 7–10. [CrossRef]
14. Haller, S. The Things in the Internet of Things. In Proceedings of the Internet of Things 2010, Tokyo, Japan, 1 December 2010.
15. Bello, O.; Zeadally, S.; Member, S. Intelligent Device-to-Device Communication in the Internet of Things. *IEEE Syst. J.* **2016**, *10*, 1172–1182. [CrossRef]
16. Pitatzis, S.; Drosos, N.; Goumopoulos, C.; Kameas, A. AmIoT: A Microservices-based IoT Platform to Orchestrate AmI Environments. In Proceedings of the 2020 16th International Conference on Intelligent Environments (IE), Madrid, Spain, 20–23 July 2020; pp. 21–28. [CrossRef]
17. Chan, M.; Campo, E.; Estève, D.; Fourniols, J.Y. Maturitas Smart homes—Current features and future perspectives. *Maturitas* **2009**, *64*, 90–97. [CrossRef] [PubMed]
18. Giffinger, R.; Fertner, C.; Kramar, H.; Meijers, E. City-ranking of European Medium-Sized Cities. pp. 1–12. Available online: http://www.smart-cities.eu/download/city_ranking_final.pdf (accessed on 16 September 2021).
19. Council, N.R.D. What Are Smarter Cities? Available online: <http://smartercities.nrdc.org/> (accessed on 16 September 2021).
20. Patrice, R. Creating “The Smart City”. 2008. Available online: https://books.google.co.jp/books/about/Creating_the_Smart_City.html?id=UzjADAEACAAJ&redir_esc=y (accessed on 16 September 2021).
21. Akande, A.; Cabral, P.; Gomes, P.; Casteleyn, S. The Lisbon ranking for smart sustainable cities in Europe. *Sustain. Cities Soc.* **2019**, *44*, 475–487. [CrossRef]
22. United Nations. *Population Distribution, Urbanization, Internal Migration and Development: An International Perspective*; United Nations: New York, NY, USA, 2011.
23. Bibri, S.E. The IoT for smart sustainable cities of the future: An analytical framework for sensor-based big data applications for environmental sustainability. *Sustain. Cities Soc.* **2018**, *38*, 230–253. [CrossRef]
24. Kummitha, R.K.R.; Crutzen, N. How do we understand smart cities? An evolutionary perspective. *Cities* **2017**, *67*, 43–52. [CrossRef]
25. González García, C.; Espada, J.P.; Valdez, E.R.N.; García-Díaz, V. Midgar: Domain-Specific Language to Generate Smart Objects for an Internet of Things Platform. In Proceedings of the 2014 Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, Birmingham, UK, 2–4 July 2014; pp. 352–357. [CrossRef]
26. Garcia, C.G.; Meana-Llorian, D.; Garcia-Diaz, V.; Jimenez, A.C.; Anzola, J.P. Midgar: Creation of a Graphic Domain-Specific Language to Generate Smart Objects for Internet of Things Scenarios Using Model-Driven Engineering. *IEEE Access* **2020**, *8*, 141872–141894. [CrossRef]
27. García, C.G.; G-Bustelo, B.C.P.; Espada, J.P.; Cueva-Fernandez, G. Midgar: Generation of heterogeneous objects interconnecting applications. A Domain Specific Language proposal for Internet of Things scenarios. *Comput. Netw.* **2014**, *64*, 143–158. [CrossRef]
28. González García, C.; Zhao, L.; García-Díaz, V. A User-Oriented Language for Specifying Interconnections Between Heterogeneous Objects in the Internet of Things. *IEEE Internet Things J.* **2019**, *6*, 3806–3819. [CrossRef]
29. García Díaz, V.; Fernandez, H.; Palacios-González, E.; Pelayo García-Bustelo, B.; Sanjuán, O.; Cueva Lovelle, J. Automated code generation support for BI with MDA TALISMAN. *Int. J. Artif. Intell. Interact. Multimed.* **2009**, *1*, 87–93.
30. Kolovos, D.S.; Paige, R.F.; Kelly, T.; Polack, F.A.C. Requirements for Domain-Specific Languages. pp. 1–4. Available online: http://www-users.cs.york.ac.uk/~tpk/req_dsls.pdf (accessed on 16 September 2021).
31. Mernik, M.; Heering, J.A.N.; Sloane, A.M. When and How to Develop Domain-Specific Languages AND. *Acm Comput. Surv.* **2005**, *37*, 316–344. [CrossRef]
32. Deursen, A.V.; Klint, P.; Visser, J. Domain-Specific Languages: An Annotated Bibliography. *ACM Sigplan Not.* **2000**, *35*, 26–36. [CrossRef]

33. Washburn, D.; Sindhu, U.; Balaouras, S.; Dines, R.A.; Hayes, N.M.; Nelson, L.E. Helping CIOs Understand “Smart City” Initiatives. 2010. Available online: https://s3-us-west-2.amazonaws.com/itworldcanada/archive/Themes/Hubs/Brainstorm/forrester_help_cios_smart_city.pdf (accessed on 16 September 2021).
34. Science, R.; Studies, M. Smart Cities Ranking of European Medium-Sized Cities. 2007. Available online: http://www.smart-cities.eu/download/smart_cities_final_report.pdf (accessed on 16 September 2021).
35. Vienna University of Technology. *European Smart Cities*; Vienna University of Technology: Vienna, Austria, 2015.
36. Nam, T.; Pardo, T.A. Conceptualizing Smart City with Dimensions of Technology, People, and Institutions. In Proceedings of the 12th Annual International Digital Government Research Conference: Digital Government Innovation in Challenging Times, College Park, MD, USA, 12–15 June 2011; pp. 282–291. [CrossRef]
37. Selic, B. Model-driven development of real-time software using OMG standards. In Proceedings of the Sixth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing, Hokkaido, Japan, 16 May 2003; pp. 4–6. [CrossRef]
38. Hailpern, B.; Tarr, P. Model-driven development: The good, the bad, and the ugly. *IBM Syst. J.* **2006**, *45*, 451–461. [CrossRef]
39. Schmidt, D.C. Guest Editor’s Introduction: Model-Driven Engineering. *Computer* **2006**, *39*, 25–31. [CrossRef]
40. Deursen, A.V.; Klint, P. Domain-Specific Language Design Requires Feature Descriptions. *J. Comput. Inf. Technol.* **2002**, *10*, 1–17. [CrossRef]
41. Mirri, S.; Prandi, C.; Salomoni, P.; Callegati, F.; Campi, A. On combining crowdsourcing, sensing and open data for an accessible smart city. In Proceedings of the 2014 8th International Conference on Next Generation Mobile Applications, Services and Technologies, NGMAST 2014, Oxford, UK, 10–12 September 2014; pp. 294–299. [CrossRef]
42. Aubry, E.; Silverston, T.; Lahmadi, A.; Festor, O. CrowdOut: A mobile crowdsourcing service for road safety in digital cities. In Proceedings of the 2014 IEEE International Conference on Pervasive Computing and Communication Workshops, PERCOM WORKSHOPS 2014, Budapest, Hungary, 24–28 March 2014; pp. 86–91. [CrossRef]
43. OMG. About the XML Metadata Interchange Specification Version 2.5.1. 2015. Available online: <https://www.omg.org/spec/XMI/2.5.1/About-XMI/> (accessed on 16 September 2021).
44. Groves, R.; Fowler, F.; Couper, M.; Lepkowski, J.; Singer, E.; Tourangeau, R. *Survey Methodology*, 2nd ed.; Wiley-Interscience: New York, NY, USA, 2009; p. 496.
45. Likert, R. A technique for the measurement of attitudes. *Arch. Psychol.* **1932**. Available online: <https://psycnet.apa.org/record/1933-01885-001> (accessed on 16 September 2021).