information

MDPI

Article

Developing Knowledge-Based Citizen Participation Platform to Support Smart City Decision Making: The Smarticipate Case Study

Zaheer Khan [1,*], Jens Dambruch [2], Jan Peters-Anders [3], Andreas Sackl [4], Anton Strasser [4], Peter Fröhlich [4], Simon Templer [5] and Kamran Soomro [1]

[1] Department of Computer Science and Creative Technologies, University of the West of England, Bristol BS16 1QY, UK; Kamran.Soomro@uwe.ac.uk

[2] Fraunhofer IGD, Competence Centre for Spatial Information Management, Fraunhoferstrasse 5, 64283 Darmstadt, Germany; Jens.Dambruch@igd.fraunhofer.de

[3] Austrian Institute of Technology-AIT, Center for Energy, Giefinggasse 6, 1210 Vienna, Austria; jan.peters-anders@ait.ac.at

[4] Austrian Institute of Technology-AIT, Center for Technology Experience, Giefinggasse 2, 1210 Vienna, Austria; Andreas.Sackl@ait.ac.at (A.S.); Anton.Strasser.fl@ait.ac.at (A.S.); peter.froehlich@ait.ac.at (P.F.)

[5] Wetransform GmbH, Fraunhoferstrasse 5, 64283 Darmstadt, Germany; simon@wetransform.to

* Correspondence: Zaheer2.Khan@uwe.ac.uk; Tel.: +44-117-328-7216

Academic Editor: Willy Susilo
Received: 28 February 2017; Accepted: 15 April 2017; Published: 21 April 2017

Abstract: Citizen participation for social innovation and co-creating urban regeneration proposals can be greatly facilitated by innovative IT systems. Such systems can use Open Government Data, visualise urban proposals in 3D models and provide automated feedback on the feasibility of the proposals. Using such a system as a communication platform between citizens and city administrations provides an integrated top-down and bottom-up urban planning and decision-making approach to smart cities. However, generating automated feedback on citizens' proposals requires modelling domain-specific knowledge i.e., vocabulary and rules, which can be applied on spatial and temporal 3D models. This paper presents the European Commission funded H2020 smarticipate project that aims to achieve the above challenge by applying it on three smart cities: Hamburg, Rome and RBKC-London. Whilst the proposed system architecture indicates various innovative features, a proof of concept of the automated feedback feature for the Hamburg use case 'planting trees' is demonstrated. Early results and lessons learned show that it is feasible to provide automated feedback on citizen-initiated proposals on specific topics. However, it is not straightforward to generalise this feature to cover more complex concepts and conditions which require specifying comprehensive domain languages, rules and appropriate tools to process them. This paper also highlights the strengths of the smarticipate platform, discusses challenges to realise its different features and suggests potential solutions.

Keywords: citizen participation; knowledge generation; automated feedback; planning proposals; domain vocabulary and rule languages

1. Introduction

Citizen participation in urban decision making is not new [1]. Emergence of Information and Communication Technologies (ICT) has transformed traditional top-down approaches (e.g., public meetings or consultations) by providing new web-based IT tools that enable citizens to take part in a participatory city planning process [2,3]. However, many current participatory tools are mainly providing commenting or voting mechanisms on the possible options of a planning proposal provided by city administrations. On the one hand, such tools greatly improve the capability of a city administration to communicate top-down plans with citizens and seek their opinion to legitimize planning decisions. On the other hand, such tools hardly support bottom-up planning or decision-making to promote co-creation and open innovation in city planning that can generate data-driven evidence-based policy making [4].

This suggests the need for participatory planning tools, which can support both top-down and bottom-up approaches, for example, allowing citizens to create new innovative ideas or proposals and facilitate dialogue between citizens and their city administration. Further, such tools should be able to make use of Open Government Data (OGD) [5] and provide contextual [6] information that may be associated with a specific location or geo-coordinates. As a result, real-time data analytics can be performed to generate new knowledge (e.g., feasibility feedback) on citizen initiated proposals for a specific topic. This increases awareness about those proposals amongst other citizens and allows them to contribute to the proposals before submitting them as a formal planning application. Existing participatory approaches often lack such a feedback feature. In addition, achieving this objective is not straightforward due to the following reasons:

(i) Need for domain knowledge which can derive rules to process proposals and generate feedback.
(ii) Enable citizens to interact with the system to create new proposals and obtain automated system generated feedback.
(iii) Fine-granularity of spatial-temporal data, format compatibility and accessibility of OGD to create and process proposals.
(iv) Visualisation of proposals in 3D landscape view.
(v) Ability to run tools from multiple platforms i.e., web, tablets and smartphones.
(vi) Need for an extensible system architecture and design to add and develop new features.

The Horizon-2020 smarticipate project [7] responds to the above research challenges by developing a smarticipate service platform for three European Cities: Hamburg (Germany), Rome (Italy) and Royal Borough of Kensington and Chelsea-London (UK). Using the smarticipate service platform, different stakeholders including citizens can interact with the system to initiate new proposals using 3D city models and obtain automated feedback on any proposed changes. The platform provides a carefully selected list of features, which are derived from the case study city requirements. These features enhance the ability of citizens to co-create, collaborate and participate in city decision making. However, these features require extensive research to provide accurate and contextual information that can enhance the effectiveness and efficiency of citizen participation in participatory planning processes. In this paper, we present a smarticipate development process, proposed platform architecture and a selected use case to highlight challenges in processing citizens' proposals and generating automated feedback.

The remainder of this paper is as follows: Section 2 covers related work followed by a brief introduction to the smarticipate project objectives in Section 3. This section also covers smarticipate system architecture and its features, followed by a selected use case that is used to develop a proof of concept to demonstrate automated feedback feature in Section 4. In Section 5, discussion and lessons learned about technical feasibility and challenges are presented. Finally, conclusions and future research directions are presented in Section 6.

2. Related Work

Our previous work on participatory governance [2,8] provides scientific review of citizen participation theories and practices in selected smart cities. Berntzen and Johannessen [9] highlighted that citizen's role in the participatory process and the competence, local knowledge and awareness of issues can produce better plans and services. In addition, their capabilities as data sensors can facilitate building liveable environments and smart cities. However, with existing known urban challenges has arisen the agenda of open governance and co-production of urban solutions [10]. The new changing landscape of ICT-enabled integrated and bottom-up participatory urban governance is driving expectations of a more effective policy implementation supported by the new legitimacy of the stakeholder coalition and the political capital of the community [11]. The interplay of social and technological innovation is transforming governance of cities, as communities expect more active engagement in the planning of their communities and the visioning of the future of their city. The traditional expert master planning is transforming towards bottom-up community and neighbourhood planning to help small communities solve big societal challenges [12,13]. The dynamic of social and technological innovation is defining a new smart city governance, addressing the complex challenges of urban planning and governance and simultaneously transforming the city governance model in fundamental ways [14]. However, only inclusive and active participation from different user groups can creatively identify and co-create urban proposals to transform local neighbourhoods.

In the above context, the smarticipate project is going beyond top-down citizen participation and encourages bottom-up participatory initiatives, which can be considered as a stepping stone towards higher levels in Arnstein's participation ladder (i.e., partnerships, delegated power and citizen control) [1]. Smarticipate also fits nicely to Participatory Method Ladder proposed by Berman [15] where he emphasises on the approaches and method to incorporate residents' perspective and needs into planning and ascend the level of citizen participation in planning processes. Among others, Beebeejaun [16] highlights one key challenge about the limited evidence demonstrating public opinion influencing the decision-making processes. In smarticipate, the openness of citizens' opinions and alternative proposal ratings provide a transparent and evidence-based approach to reflect citizens needs to influence planning decisions.

Smarticipate reuses results from the urbanAPI project [2], where 3D virtual planning tools were developed and tested with domain experts. Dambruch and Krämer [3] report about how such tools could be used in public participation processes. Smarticipate takes on board these findings and goes beyond visualisation of planning proposals by including interactive feedback mechanisms in 2D and 3D visual models. Similarly, Ruppert et al. [17] provide an overview of visual decision-making support for policy making by demonstrating one of the urbanAPI project [2] case studies on eParticipation in urban planning. They conclude that visual technologies are useful for communication and support a dialogue from experts to citizens. These conclusions provide the basis to use 2D and 3D visualisation of proposals in the smarticipate platform so that a dialogue from citizens to experts can be initiated. Krämer et al. [18,19] suggested to use Domain-Specific Language (DSL) to define rules which can be used to define constraints and domain knowledge to be applied in a policy cycle. This provides the basis to use DSL to design and develop the automated feedback feature.

In addition to above, there are several participatory projects in urban planning which mostly focus on a specific topic or features to support the planning process [20]. In the smarticipate context, the most relevant initiatives around the world are:

- **CiviQ** (http://www.civiq.eu/, last accessed: 27 February 2017)—It provides visual services that visualise the flow of all stakeholder's opinions, from submissions, consultation and deliberation process;
- **COLAB** (http://www.colab.re/, last accessed: 27 February 2017)—Is a citizen-to-government engagement platform for issue reporting and public services evaluation as well as participation in decision-making process;
- **COMMONPLACE** (http://www.commonplace.is/, last accessed: 27 February 2017)—Is an online consultation platform for local participation through dialogues for making more compelling proposals with real-time feedback and analysis presented through live dashboard;
- **Sticky World** (http://info.stickyworld.com/, last accessed: 27 February 2017)—Allows people to upload and share different types of contents including multimedia contents e.g., videos, images, pdf, audios to exchange ideas visually with others so that they can add comments, while backend systems also generate participative statistics;

- **City Swipe** (http://www.dtsmcityswipe.com/, last accessed: 27 February 2017)—Provides an intuitive citizen engagement platform to learn about citizens' preferences and concerns about the city's urban core through quick questions, which is then used in long-term city plan;
- **Land Sight** (http://www.landinsight.io/, last accessed: 27 February 2017)—A set of tools that can provide information about a selected site/land, use of the site and can perform preliminary viability assessment indicating what kind of development is likely to receive planning permission;
- **Flux Metro** (https://metro.flux.io/metro/, last accessed: 27 February 2017)—Allows users to interact with a sample area of Austin, Texas, in 3D environment to visualise a site's information e.g., context and constraints, heights and shadows. It can be used to get information about building plots and parcels by combining data from different sources which is becoming a challenge to understand development potential and predict profitability outcomes in given scenarios;
- **City Life Management** (http://www.siemens.com/innovation/en/home/pictures-of-the-future/infrastructure-and-finance/livable-and-sustainable-cities-virtual-urban-planning.html, last accessed: 27 February 2017)—Is an online engagement platform that can help to calculate short- and long-term impact of a planning intervention. For example, a user can place a building object in a 3D model of a site and the platform can generate results such as energy demands, impact on traffic flows, air pollution, etc.;
- **UrbanPlanAR** (http://urbanplanar.com/, last accessed: 27 February 2017)—Attempts to use mobile augmented reality to capture real-time-in-field visualisation of a proposed development at a site using 3D data to visualise development potential;
- **Piazza platform** (http://www.piazza.eu/, last accessed: 12 April 2017)—Provides a digital platform to facilitate dialogue between citizens and city administration to test new urban infrastructure or services before entering the planning or implementation phase.

All the above initiatives cover different aspects of participatory governance. However, most of them are either concentrating on visualisation or communication or planning and expected impact. To the best of our knowledge, no one existing solution fully supports both top-down as well as bottom-up citizen engagement and provides features such as 2D/3D visualisation; change in proposal and obtaining automated feedback; dialogue exchange; citizen communication; preference selection; alternative proposals to support evidence-based urban plans. This makes smarticipate unique and takes it beyond the state-of-the-art.

3. The Smarticipate Platform

The smarticipate project [7] aims to develop an online participatory platform that is accessible through PC/web, tablets and smartphones. The objective of the project is to enable citizens and city administration to establish a dialogue on new planning proposals or services. The objectives here are to (i) make effective use of OGD; (ii) get citizens opinion on the proposed planning initiatives by city administration—promoting top-down participatory planning; and (iii) to enable citizens to co-create and share new innovative proposals with the community, thus promoting bottom-up planning and open innovation [21]. Citizens can create their proposals. Others can interact with those proposals and the automated feedback feature of the platform provides impact assessment e.g., feasibility details about a proposal when certain urban infrastructural parameters are changed. For example, feedback can be "whether a proposal is compliant to local planning regulations?" or "what is the budget or cost associated with the proposal?" etc.

3.1. Smarticipate Methodology

Like a typical system development process, Figure 1 depicts that smarticipate starts with identification of use cases and requirements from case study cities: Hamburg, Rome and RBKC-London. The CoReS method [22] was applied and a number of requirements were gathered, analysed and validated. For requirements gathering, three requirements gathering workshops were organised with case study cities. Selected use cases and requirements were defined. Each requirement statement consists of description, rationale, owner, acceptance criteria, validation status and level of importance. All requirements are managed through an online collaborative project management system, Redmine [23]. This enabled city stakeholders to refine, update and validate these requirements. There was a total of six use cases and 72 requirements.

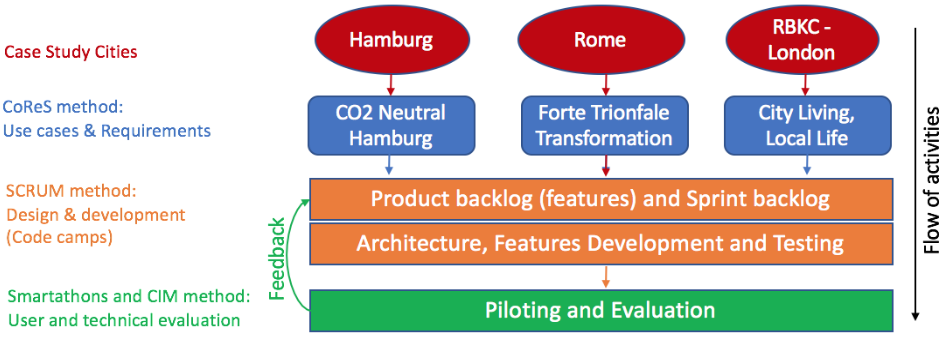
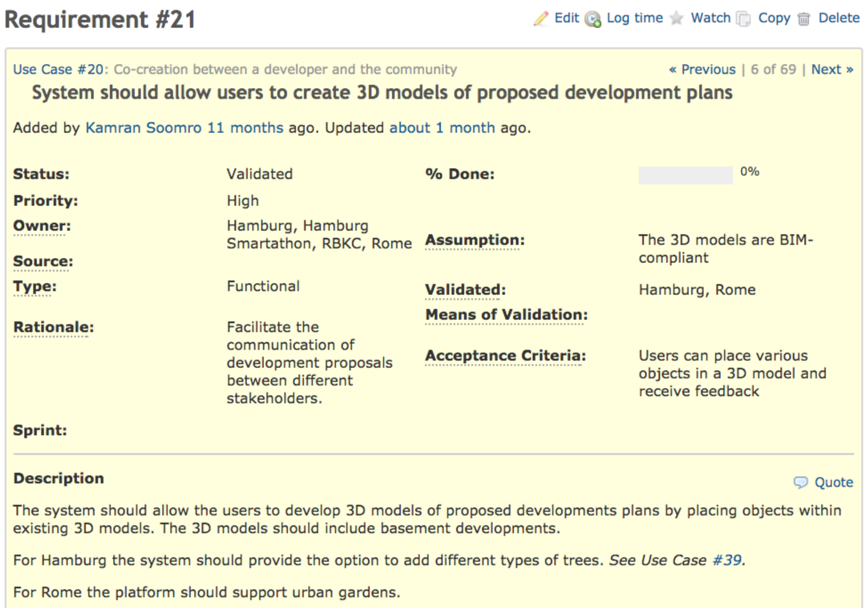


Figure 1. Depicts the overall smarticipate system development methodology.

Then SCRUM methodology [24] is used for the design and development of specific required features. As a result, a product backlog and sprint backlogs were created. The objective was that these features are tested and validated by end users from case study cities so that the smarticipate platform can be deployed and evaluated in a real environment. A total of 117 features were derived from the requirements and inserted in the Redmine requirements management system. As an example, please see Figures 2 and 3, which depict one example of a requirement and its associated feature.



Requirement #21

Use Case #20: Co-creation between a developer and the community

System should allow users to create 3D models of proposed development plans

Added by Kamran Soomro 11 months ago. Updated about 1 month ago.

Status:	Validated	% Done:	0%
Priority:	High		
Owner:	Hamburg, Hamburg Smartathon, RBKC, Rome	Assumption:	The 3D models are BIM-compliant
Source:			
Type:	Functional	Validated:	Hamburg, Rome
		Means of Validation:	
Rationale:	Facilitate the communication of development proposals between different stakeholders.	Acceptance Criteria:	Users can place various objects in a 3D model and receive feedback
Sprint:			

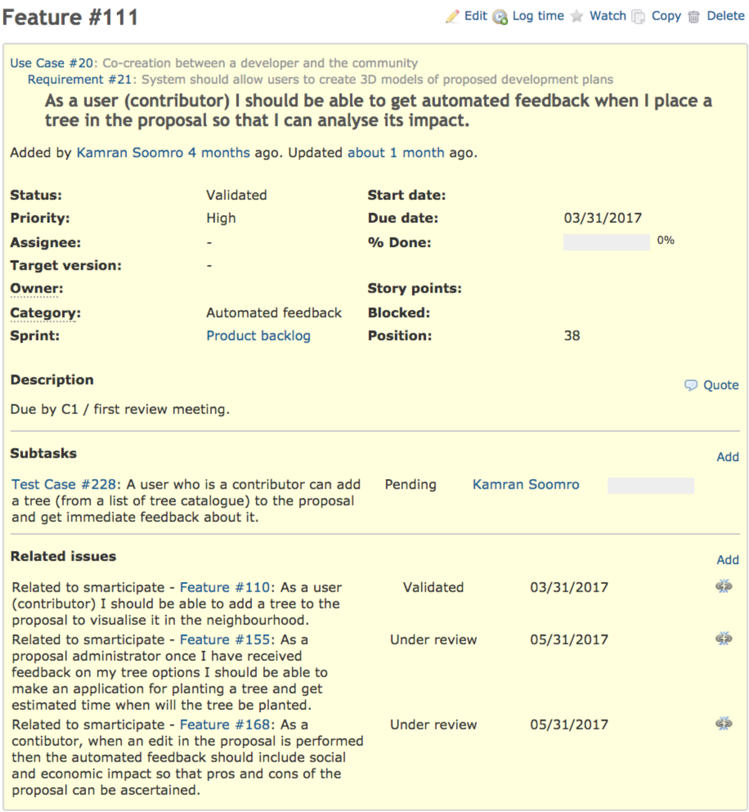
Description

The system should allow the users to develop 3D models of proposed developments plans by placing objects within existing 3D models. The 3D models should include basement developments.

For Hamburg the system should provide the option to add different types of trees. *See Use Case #39*.

For Rome the platform should support urban gardens.

Figure 2. Smarticipate requirements management.



Feature #111

Use Case #20: Co-creation between a developer and the community

Requirement #21: System should allow users to create 3D models of proposed development plans

As a user (contributor) I should be able to get automated feedback when I place a tree in the proposal so that I can analyse its impact.

Added by Kamran Soomro 4 months ago. Updated about 1 month ago.

Status:	Validated	**Start date:**	
Priority:	High	**Due date:**	03/31/2017
Assignee:	-	**% Done:**	0%
Target version:	-		
Owner:		**Story points:**	
Category:	Automated feedback	**Blocked:**	
Sprint:	Product backlog	**Position:**	38

Description

Due by C1 / first review meeting.

Subtasks

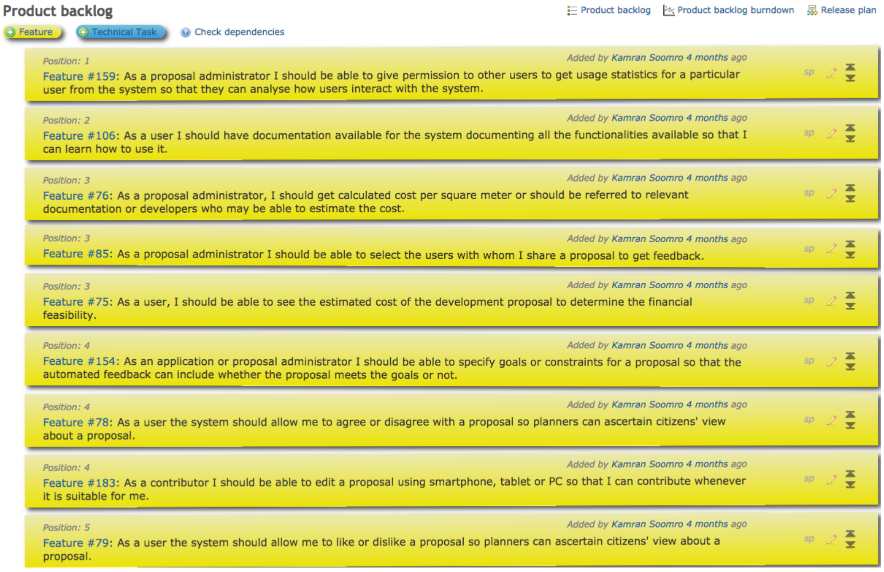
Test Case #228: A user who is a contributor can add a tree (from a list of tree catalogue) to the proposal and get immediate feedback about it.	Pending	Kamran Soomro	

Related issues

Related to smarticipate - Feature #110: As a user (contributor) I should be able to add a tree to the proposal to visualise it in the neighbourhood.	Validated	03/31/2017
Related to smarticipate - Feature #155: As a proposal administrator once I have received feedback on my tree options I should be able to make an application for planting a tree and get estimated time when will the tree be planted.	Under review	05/31/2017
Related to smarticipate - Feature #168: As a contibutor, when an edit in the proposal is performed then the automated feedback should include social and economic impact so that pros and cons of the proposal can be ascertained.	Under review	05/31/2017

Figure 3. An example of a feature derived from a requirement.

As SCRUM methodology is applied for the development of the smarticipate platform, product backlog and sprint backlogs are also managed in the Redmine requirements management system. Figures 4 and 5 depict a product backlog with high priority features and sprint boards. Hence, all requirements, features, development tasks and test cases are managed in one content management system that provides forward and backward traceability and an up-to-date status of feature development.



Product backlog

Position: 1 Added by Kamran Soomro 4 months ago
Feature #159: As a proposal administrator I should be able to give permission to other users to get usage statistics for a particular user from the system so that they can analyse how users interact with the system.

Position: 2 Added by Kamran Soomro 4 months ago
Feature #106: As a user I should have documentation available for the system documenting all the functionalities available so that I can learn how to use it.

Position: 3 Added by Kamran Soomro 4 months ago
Feature #76: As a proposal administrator, I should get calculated cost per square meter or should be referred to relevant documentation or developers who may be able to estimate the cost.

Position: 3 Added by Kamran Soomro 4 months ago
Feature #85: As a proposal administrator I should be able to select the users with whom I share a proposal to get feedback.

Position: 3 Added by Kamran Soomro 4 months ago
Feature #75: As a user, I should be able to see the estimated cost of the development proposal to determine the financial feasibility.

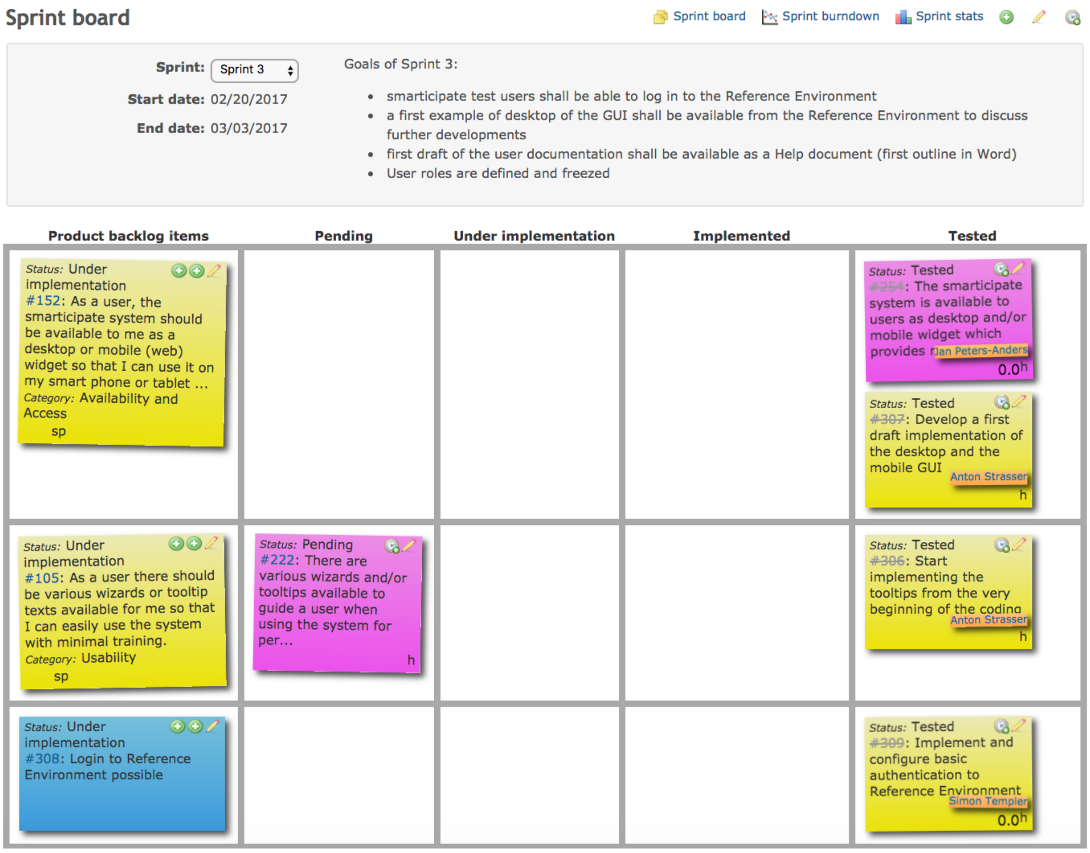
Position: 4 Added by Kamran Soomro 4 months ago
Feature #154: As an application or proposal administrator I should be able to specify goals or constraints for a proposal so that the automated feedback can include whether the proposal meets the goals or not.

Position: 4 Added by Kamran Soomro 4 months ago
Feature #78: As a user the system should allow me to agree or disagree with a proposal so planners can ascertain citizens' view about a proposal.

Position: 4 Added by Kamran Soomro 4 months ago
Feature #183: As a contributor I should be able to edit a proposal using smartphone, tablet or PC so that I can contribute whenever it is suitable for me.

Position: 5 Added by Kamran Soomro 4 months ago
Feature #79: As a user the system should allow me to like or dislike a proposal so planners can ascertain citizens' view about a proposal.

Figure 4. Product backlog—(total 117 features).



Sprint board

Sprint: Sprint 3

Start date: 02/20/2017

End date: 03/03/2017

Goals of Sprint 3:

- smarticipate test users shall be able to log in to the Reference Environment
- a first example of desktop of the GUI shall be available from the Reference Environment to discuss further developments
- first draft of the user documentation shall be available as a Help document (first outline in Word)
- User roles are defined and freezed

Product backlog items	Pending	Under implementation	Implemented	Tested
Status: Under implementation #152: As a user, the smarticipate system should be available to me as a desktop or mobile (web) widget so that I can use it on my smart phone or tablet ... *Category:* Availability and Access sp				*Status:* Tested ~~#254~~: The smarticipate system is available to users as desktop and/or mobile widget which provides r Jan Peters-Anders 0.0h *Status:* Tested ~~#307~~: Develop a first draft implementation of the desktop and the mobile GUI Anton Strasser h
Status: Under implementation #105: As a user there should be various wizards or tooltip texts available for me so that I can easily use the system with minimal training. *Category:* Usability sp	*Status:* Pending #222: There are various wizards and/or tooltips available to guide a user when using the system for per... h			*Status:* Tested ~~#306~~: Start implementing the tooltips from the very beginning of the coding Anton Strasser h
Status: Under implementation #308: Login to Reference Environment possible				*Status:* Tested ~~#309~~: Implement and configure basic authentication to Reference Environment Simon Templer 0.0h

Figure 5. Sprint board—SCRUM sprint planning (colouring scheme shows different categories of items e.g., yellow colours are features, blue colours are technical tasks, purple colours are test cases).

For piloting, Smartathons (a term derived from hackathons) were organised to gather citizens' requirements in co-designing the smarticipate platform. These Smartathons generated new requirements which were also added in the Redmine requirements management system. For evaluation, a Criteria Indicators and Metrics (CIM) approach [25,26] is applied. Using the CIM approach, test cases are defined for each requirement. These test cases are also added in the Redmine system and are used by developers to test the features, as depicted in Figure 5. In addition, both online and In-system evaluation techniques are planned for user-based evaluation exercises to acquire feedback and improve smarticipate features.

3.2. A Selected Use Case: CO_2 Neutral Hamburg—Tree Planting

To demonstrate smarticipate platform usefulness, one of the use cases from Hamburg is selected. This selected use case (i) shows the relevance of the smarticipate project for citizen participation and informed choices when creating proposals; and (ii) develops a proof of concept to demonstrate selected smarticipate platform features.

Based on Hamburg Transparency Act, Hamburg's open data portal provides a huge amount of OGD. However, it is not straightforward for a non-IT person to make effective use of this data. Hamburg would like citizens to make effective use of OGD for their informed decision making. In this respect, the CO_2 Neutral Hamburg use case aims to enable its citizens to make informed choices about tree plantation in their neighbourhoods. This means that a citizen should be able to select a location for tree plantation through the smarticipate platform. This tree may belong to some specific species. At the moment, without manual expert intervention, it is not possible to obtain useful information on the feasibility of tree plantation at the selected location and share it with other citizens. Through smarticipate, a citizen not only will be able to select a location for tree plantation proposal but also would be able to get analytical feedback. This feedback may include (i) whether or not it is feasible to plant a tree at the selected location? (ii) If it is not possible then why is it not possible? And suggest an alternative location that is more feasible. (iii) What is the budget and cost of tree plantation? (iv) What is the expected environmental impact i.e., CO_2 reduction, etc. Based on this information, that citizen can share his/her proposal with the community to obtain suggestions and assess any social impact. This will help that citizen to decide whether to go ahead with the tree planting application with city administration or share with other stakeholders for fund raising (i.e., crowd funding). Hence, the smarticipate platform will enable citizens to make use of OGD, visualise tree plantation, obtain feedback on tree plantation, communicate with city administration and other citizens for their opinions.

3.3. Smarticipate System Architecture

The smarticipate platform is designed as a responsive web application (i.e., responsive web design) so that it can be available on different devices and screen sizes and accessible to anyone who is interested in participating in planning proposals. Figure 6 depicts the overall smarticipate system architecture.

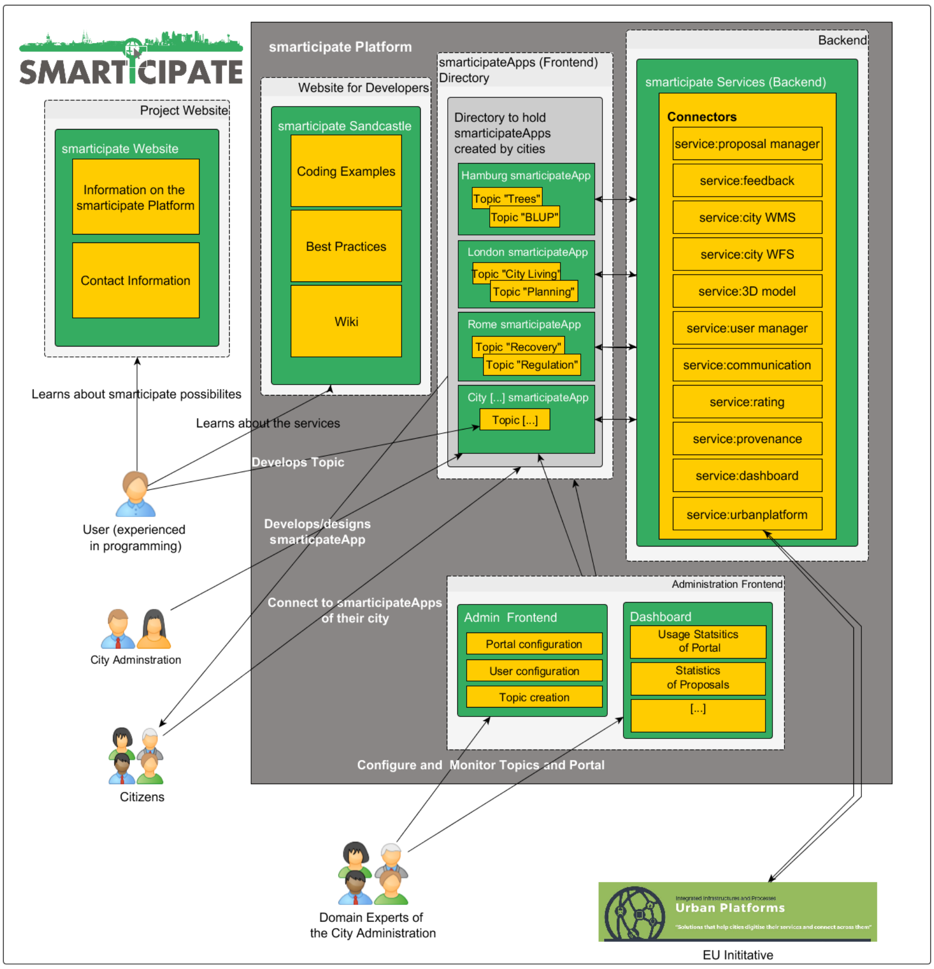


Figure 6. Smarticipate system architecture.

The overall idea here is to have the smarticipate website component (on the left-hand side in Figure 6) serving as an entry point to the platform. During the lifetime of smarticipate, the project is running its own Docker server for continuous testing, co-designing features and features demonstration purposes.

The features of the smarticipate platform are (Figure 6, grey box):

- **The Website for developers**: It serves as an information hub for users able to code or write software programs. Here, they find coding examples and best practices which enable them to code a Topic for the smarticipateApp of a city e.g., building refurbishments, brown field regeneration, etc.
- **SmarticipateApp(s)**: This is the directory which holds the smarticipateApp of one city (e.g., dedicated server) or more cities (e.g., deployment through centralised server on cloud). The apps themselves contain different Topics.
- **Topic(s)**: Topics hold the content of a participation process being tackled by the city (e.g., the tree planting topic of Hamburg, see below Figure 7). The smarticipateApp comprises of one or more Topics. These topics can be built using OGC or data from other sources which are not available in the public domain.
- **Backend-smarticipate services**: This is the backend of the platform which connects to the different data sources needed for the participation Topic (e.g., OGD) as well as connections to social media and configuration services for the actual smarticipate platform instance. These services provide functionality for different required features. Currently, the following services are included. However, additional services can be developed and linked due to a micro-services approach (discussed later).

    - **Proposal manager service:** This service facilitates the different proposals suggested in a specific Topic of the platform. It also manages the different comments and edits made by the different users of the platform.
    - **Feedback service:** This is one of the core services of the smarticipate platform. It provides direct feedback for each Topic (from pre-configured data layers and rules). The feedback is requested by clicking on a location or object within the map and the user is then provided with information if a proposal is feasible at this location and if not, why this is the case. In this way, the city planning procedures become more transparent for the citizens.
    - **3D Model service:** This service provides 3D data for visualisation purposes (if the city has such data available).
    - **WFS (Web Feature Service):** This service provides a WFS interface to the platform's Geoserver. If a certain dataset is not available as Open Data, citizens can create their own service to be used in their Topics.
    - **WMS (Web Map Service):** Like the WFS service, this service provides a WMS interface to the platform's Geoserver which enables the creation of own (visual) map layers for the new Topics. This service is useful when required map data is not available via Open Government Data portals.
    - **User manager service:** This service provides the interface to the user and user rights management system of the platform. This means that different roles and permissions can be set up by the platform administrator.
    - **Communication or notification service:** This service manages the communication between the users of the platform and sends notifications about new Topics and proposals.
    - **Rating service:** This service manages the ratings/voting of the different proposals done by the users of the platform.
    - **Provenance service:** This service is used to keep a track record of all proposals and their edits, enabling a later analysis of the process, which can then lead to a change in the future proceedings of the city planning.
    - **Dashboard service:** This service provides the interface to the data that is needed to generate various statistics. It provides statistical information such as latest logins of users, number of proposals for each Topic, number of edits, number of votes in favour of a proposal etc.

- ○ **Urban platform service:** This service will enable a connection to the Urban Platform initiative's server. This will ideally be a generic service which will allow similar platforms to be connected in the future.

- **Administration Frontend**: This is the frontend to configure and monitor the different Topics and their users involved.
- **System Dashboard:** This is the visual interface which provides all statistics about topics and associated proposals e.g., trend analysis, rating of each proposal, number of users participating in a topic, etc.

Figure 6 also depicts different roles (avatars), which interact with different platform services. These roles are:

- **User (experienced in programming):** A person with experience in programming who can create Topics using the smarticipate API.
- **City Administration:** Represents the users within the city administration who configure and monitor the smarticipate platform.
- **Citizens:** Refers to the users who use the smarticipate platform to give their opinion to the smarticipate Topics and Proposals.
- **Domain Experts:** These can be urban planners, builders, infrastructure developers, local businesses who are able to share their expert knowledge.
- **External EU Initiatives:** Other projects and initiatives such as Urban Platforms.

The project's scenarios are represented by their logic in the frontend and the backend of the smarticipate platform. The backend of the smarticipate platform has been designed to be as generic (i.e., reproducible) as possible. This will allow other cities/citizens to make use of smarticipate interfaces to create their topics of interest. The following Figure 7 depicts the flow of activities of the selected Hamburg use case. This shows how different services work together to deliver the tree plantation use case. Figure 7 is structured as follows:

- The second column depicts the "User".
- The third column ("smarticipateApp") depicts the workflow of the frontend app to be developed.
- The columns named "service: [. . .]" depict the backend services to be consumed by the smarticipateApp.

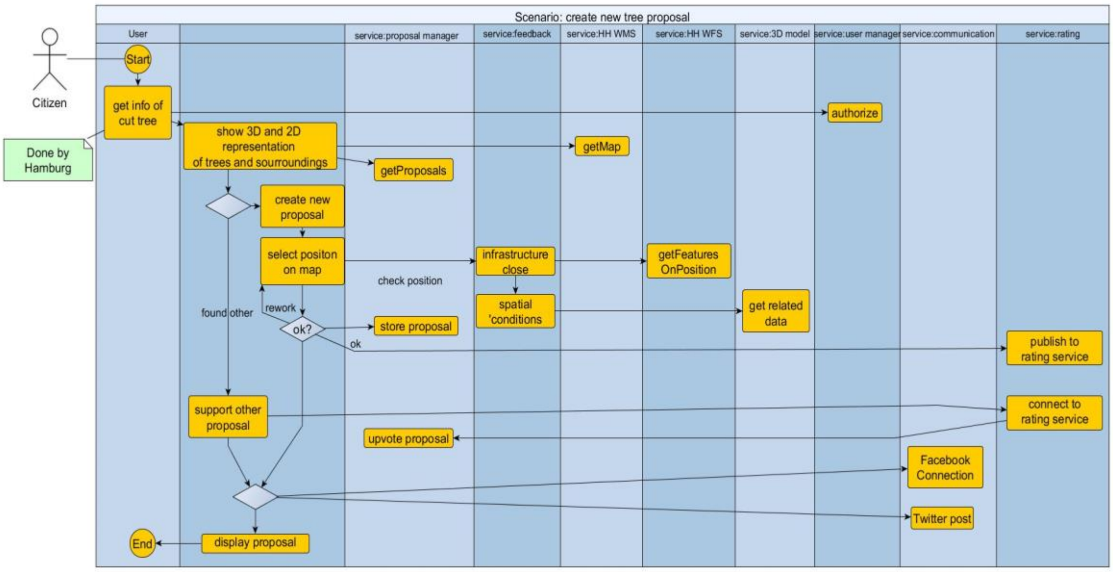


Figure 7. Example of an Activity Diagram: Hamburg: CO_2 Neutral-Tree Planning Scenario.

3.4. Implementation and Deployment Setup

This section presents the selected technologies and deployment setup details.

3.4.1. The SmarticipateApp

The smarticipateApp is using an HTML5/Javascript-based approach to create a platform-independent, responsive, and reusable user interface, which enables the delivery of high usability and accessibility to satisfy end users. As a main scenario, smarticipate sees the usage of smartphones for creating and discussing citizens' proposals.

With HTML5, it is possible to employ the Progressive Web App (PWA) (https://developers.google.com/web/progressive-web-apps/, last accessed: 27 February 2017) development model, which is the successor of so-called hybrid apps. The Web application itself is designed as a single-page-application (SPA), built upon the React JavaScript library (https://en.wikipedia.org/wiki/Single-page_application, last accessed: 27 February 2017), which follows a modern component-based approach enabling a high degree of modularization of the codebase and reusability of existing components. There exists a rich set of components, such as UI-toolkits, state-management, routing, etc., that can be used to tailor the application to the specific needs of the project. With React-Native (https://facebook.github.io/react-native/, last accessed: 27 February 2017), it is planned to create native mobile apps using React, which can perform better than progressive or hybrid apps.

3.4.2. Microservices Approach

Namiot and Sneps-Sneppe [27] describe a micro-service architecture as an approach to develop an application as a set of small independent services. Each of the services is running in its own independent environment and the services can communicate via some lightweight mechanism such as HTTP or HTTPS. An example of the micro service approach is Netflix (https://www.netflix.com/, last accessed: 23 February 2017). In the smarticipate context, it is obvious that change and adaption to users' needs is the key to success. Apart from this, the technical integration in an existing IT-infrastructure is also a common goal to avoid replicating functionality and data, which means that an open architectural approach is needed. This is suitable for the smarticipate platform where integration of smarticipate services with current IT systems of a city administration is required. For instance, an urban regeneration proposal created by a citizen can be directly submitted, with all citizen participatory and feedback evidence, to the planning department through the city planning application portal. To meet such needs, the smarticipate platform is based on Micro-Service Oriented Software Architecture. This approach provides strong isolation and loosely coupled services with a single goal so that any changes in requirements can be manageable. For example, the user management service mentioned in Figure 6 only deals with managing users, not billing users, not communicating with other users and so forth. The micro-service approach also provides flexibility in using the most appropriate tools to implement a service. For example, when web-based formats and protocols are used i.e., the technology providing the service can be selected as best suited for the case (as demonstrated in the feedback service proof of concept). This means that services using different technologies can be combined easily. Micro services are well suited for smarticipate's agile development approach since small services can be developed completely in one sprint and the common codebase is kept small.

However, the above architectural approach is not without challenges. For instance, the complexity of orchestration and process management is now located at the network level in a distributed, loosely coupled system. This means that smarticipate must expect:

- Fault tolerance e.g., network errors or outages may occur at any time.
- Latency and limited bandwidth for network access i.e., data access is not cheap.
- Network security must be ensured.
- Changes in network topology can have impacts; networks can be inhomogeneous.

- Administration of a "zoo of machines and services" is expensive.
- Testing requires more effort and additional technology for distributed systems.

3.4.3. Docker as Container Platform

Many of the issues are covered by using a lightweight virtualisation technology for the deployment of smarticipate services. In this respect, Docker-Containers (http://www.docker.com/, last accessed: 19 April 2017) are selected to provide the runtime environment for most smarticipate platform services. Docker provides a minimalistic, flexible, easy to setup and manage Linux environment for a service that works independently and isolated in a container. With these containers, it is also possible to exchange services at run time or have sophisticated Cloud Computing technology which utilises available resources more effectively. The network complexity with mapping network paths and addresses to services can be virtualised or can be provided via docker-compose scripts. These scripts describe which services cooperate and which services are needed to be deployed for runtime environments. In the simplest case, all services can run at a single developer machine. The same setup can also be deployed and run in a production environment on a reliable and redundant hardware in cloud environments without worrying about the required libraries. This gives the system ultimate flexibility to involve citizens as testers or developers. If their needs change in the future, the users of the system can migrate to cloud service providers or set up their own cloud.

4. Automated Feedback Feature: An Example Service of the Smarticipate Platform

Now we take one example service to demonstrate proof of concept. The automated feedback service is selected due to its highly unique capabilities and high demand by the end users. This proof of concept will show that the feedback service uses various technologies for the implementation of certain capabilities and the micro-service approach can be connected with the smarticipate platform. In the following sections, we will cover the conceptual design, and associated concepts such as DSL, experimental setup and proof of concept of the automated feedback service.

4.1. Conceptual Basis and Feature Design

The traditional way of participation is often not transparent for citizens and tends to be delayed in terms of evaluation of proposals by responsible officials. To obtain any feedback on citizen proposals, domain experts need to be involved and workload for such experts is often high. On the other hand, data alone published as OGD or open data is often hard to interpret for lay-people and they need at least some guidance and help from experts to understand it. The basic idea of automated feedback is that the system can process citizens' proposals and generate feedback using open or private data. Dambruch and Krämer [3] suggested an interactive 3D scenario creator where people can comment on proposals and upload their own designs in 3D using standard web technology. However, in the evaluation of the 3D scenario creator, a need for giving hints on the feasibility of the designs proposed was evident [8]. Later, Malewski, Dambruch and Krämer [28] presented a concept of a combination of 3D visualisation and interaction components with an ontology-driven rule editor based on domain-specific languages. The 3D visualisation, on the one hand, enables stakeholders to present and discuss urban plans. On the other hand, the rule editor particularly targets expert users who need to perform spatial analyses on urban data or want to configure the 3D scene according to custom rules. They use rules not only to compute results but also to create visual representations of the results, for example objects with specific metadata attributes can be automatically coloured differently. An example of such a rule would be highlighting all buildings taller than a specific height so that they can be easily identified visually. They conclude that a DSL rule editor in combination with a visualisation component offers a new way for GIS data experts to communicate their analysis process and results to non-experts. This idea has been taken up by the smarticipate automated feedback service to compute and generate automated feedback on people's planning proposals. The conceptual design of the feedback feature is based on modelling the domain knowledge and technical rules.

However, the feedback service should not be seen as a decision maker, rather it is a support tool to assess a particular proposal (e.g., infrastructure or public service) before initiating the formal planning application. The feedback feature relies on the data available and user-defined representation of policy or planning rules. There can be many cases where data is not sufficient or domain-specific or policy rules are not easily transferrable to scripts and hence manual intervention is needed.

The rules can be of two categories. First the actual rules defined by city administration or domain experts based on planning regulations or general legislation. The second category is the technical machine-readable rules defined in specific languages or scripts. In an ideal case, all actual rules could be modelled or mapped to technical rules and functions, which will result in deterministic behaviour of the feedback feature. The experience shows that, even using DSL, this is not very likely due to the way the rules have been developed over long periods and hence a pragmatic approach is needed. From a practical point of view, cities can benefit from the feedback service as it could handle rather simple to medium-complex tedious routine requests for experts, while experts can concentrate on complex and ambiguous cases.

4.1.1. Modelling Domain Knowledge

Modelling domain knowledge via Domain-Specific Languages has several advantages. There exist methodologies [29] for modelling, which have been successfully implemented and tested [18]. Using Krämer's approach [18], the domain semantics are also covered within the language and no specific data format is required, e.g., data is annotated on access via the language elements. This means that no special data annotation format such as Resource Description Framework (RDF) is needed. In addition, data is given in standard geospatial formats and existing services available in cities can be used.

Based on the above analysis, the smarticipate automated feedback service utilises DSLs due to the following reasons:

- Data availability—usable annotated data or even RDF is not available in the participating cities;
- Annotating data and/or redundant data storage is not feasible for cities;
- Expert users are typically not familiar with complex IT-concepts and would need support in representing domain-specific concepts;
- Definition of dynamic aspects, actions and visualisation is also important besides reasoning.

Deriving knowledge-based results from raw data should be easy for users with high IT skillsets. This means that there should be means for representing expert knowledge on different abstraction levels.

Several feedback workshops have been conducted in the smarticipate case study cities to derive and model the domain knowledge and define rules for automated reasoning. Domain experts e.g., urban planners, GIS experts, social housing experts etc., participated in these workshops. Using the Krämer approach [18] and textual noun–verb analysis approach, the rules were derived based on the following steps:

1. Requirements gathering and analysis.
2. Definition and analysis of Use Cases and User Stories.
3. Domain analysis.
4. Definition of a terminology and a Domain Model.
5. Mapping of terminology to software artefacts and actions.
6. Building of sample DSL scripts and transforming it into a formal grammar.
7. Review, test and reiterate if needed.

The result of this process is a formal grammar which describes the Domain-Specific Language. Scripts created based on this grammar are executed in a generic service environment, combining data from various sources to compute a result and an explanation which rules have been fulfilled or violated.

This provides the possibility to convey a clear explanation why a proposal is possible or not. Below, we present a small example of DSL for the tree plantation use case.

The analysis is performed using the noun–verb analysis technique. A thorough analysis of the use case and related documents provided a list of nouns, verbs and properties which are used to define basic concepts and possible actions. An excerpt of the results is given in Tables 1 and 2 below.

Table 1. Nouns representing concepts.

Possible Nouns (Concepts)			
Proposal	Cost	Goal	Power Lines
Infrastructure	Water	Gas	Communication
Private Land	Land use	Planned Actions	Tree
Species	Neighbourhood	Building	Street Lighting
Traffic Signs	Flooding area	Condition of soil	Shadow and Light
Sidewalk	Street Access	Bus Lane	Position

Table 2. Words representing actions.

Possible Actions			
Agree/Disagree	Calculate costs	Link	Exclude/Include
Measure distance	Determine species	Define species	Growth simulation
Flooding simulation	Intersection	Shadow masking	

4.1.2. Conceptual Modelling for Rules—An Example

An example rule from the above selected use case in plain text is:

> *"Distance to Street Lights: Trees grow and possibly will mask street lights nearby. A minimum distance should be kept from such positions. Positions of street lights need to be given."*

Analysis reveals the concepts such as *Street Light*, *Tree*, *Position* and possible actions such as *distance calculation*, *tree growth simulation*, *shadow masking* are potential candidates for domain knowledge. The basic idea is to declare a term that can be computed using the above terminology and when the evaluation of an event is positive, trigger some action which leads to rule templates such as this:

> *When <Term> then <Action> else <Action>*

Thus, the example can be phrased like this:

> *WHEN X of Tree AND NOT EXISTS Y of Lamp and distance(X, Y) greater than or equals 8m*
> *THEN RETURN failure description*

Using the above approach, a repository of domain rules can be defined. These rules work with the concepts defined in DSL and are applied on the citizens' created proposals for reasoning. As a result, any violation of these rules will notify citizens why a proposal is not feasible. This reasoning can also identify what alternatives can be used for the proposal.

The challenge here is to find an appropriate level of abstraction for the domain-specific language. On the one hand, it must be expressive enough to cater for the needs of the domain. On the other hand, it needs to be understandable and easy to use i.e., any technical details should be hidden if possible. As the language is specific for the domain, it should not be too general. However, there may be a lot of commonalities for different topics such as distance measuring, which can be generalised and therefore it may be possible to develop families of domain-specific languages suited for a specific topic. This means that languages may be similar in structure and semantics and only differ in concrete tokens or words for the same operation or concept. As a result, the assumption is that the reusability of basic language elements will be considerably high. Generally, a declarative approach is very useful as it defines what should be the outcome and not how to compute it step by step.

Also, another important aspect is the quality of data available in cities. Planning and infrastructure data may not be very accurate, e.g., for underground infrastructure such as pipes and power lines, there may be just a corridor given where the infrastructure is located, and quite often there is no depth data given. This vagueness needs to be considered and made transparent in computed results as this will make results credible for end users. For instance, knowing exact details of utility infrastructure i.e., geo-coordinates of underground pipelines, etc., can be beneficial for the 'tree planting' use case as it can help to compute feedback about why a tree cannot be planted at a selected location i.e., the proposed tree location is not suitable because it may damage the underground water pipes, fiber optic cable, etc. Therefore, the Hamburg open data portal is analysed to assess the availability of such data. Also, this can help to point out which data is useful for what purpose and should be collected in the future. The same is true for any assumptions made in proposal creation, especially when statements about the cost of a proposal are involved e.g. building refurbishment costs. Typically, there are rules of thumb and literature about calculating costs of buildings per square meter. However, applying these cost-measuring approaches in different contexts needs to be properly tested. An interesting option could also be to build a database of past planning applications where costs were calculated. This could be analysed and provide estimated costs inputs for similar proposals created in the smarticipate platform. At the moment, this is subject for future research.

The analysis was carried out for all examples elaborated in the feedback workshop and a domain model was generated on that basis (Tables 1 and 2). The purpose of the domain model is to show which concepts (and their relationships) are included and can be handled by the feedback feature service. This means that the feedback service will rely on the richness of this domain model i.e., more concepts (and their relationships) will allow more objects/concepts to be handled as part of feedback calculation. Below, we present a rule script example using DSL:

Sample Code Listing 1. A Technical Rule Script using Domain-Specific Language (DSL).

```
Data {
    User input point(wgs84) POSITION
    Map {
        Source.position = project(EPSG:25832, dest.position)
    }
    Datasource wfs TREE http://x.y.z
    Map {
        Source.name = dest.treename
        ...
    }
}
Rules {
    When exists TREE(x) and distance(x, position) less than 8m
    Then result.add("Tree too close to existing tree")
}

Result {
 . . .
}
```

A rule script (as shown in Code Listing 1) can contain several sections as detailed below:

- The ***Data*** section defines what is the input for the script, how to obtain it (e.g., via OGC WebFeatureService-WFS) and what is expected from the caller (user input e.g., geo-coordinates of a click on a map) as input.
- The ***Map*** section is optional and can be used for mapping the data to the terminology used inside the scripts or carry out geospatial re-projection on geo data.
- The ***Rules*** section contains the technical rules that contribute to the results of the service.

An implicit construct named result can be used for simple results as text to return to the caller. If this is not appropriate, an optional ***Result*** section can be used to return complex results. For example, a Map statement can serve a similar purpose as the Map statement in the Data section to map into another spatial reference system or visualise the results in a 2D or 3D format (e.g., GLTF – GL Transformation Format - https://www.khronos.org/gltf, last accessed: 18.04.2017, X3D – Extensible 3D - http://www.web3d.org/x3d/what-x3d, last accessed: 18.04.2017, GML – Geography Markup Language - http://www.opengeospatial.org/standards/gml, last accessed: 18.04.2017).

4.1.3. Conceptual Design of the Feedback Service

A dedicated service is developed that provides a framework which is responsible for managing data and network access and dispatch of DSL-programmes to generate automated feedback. Figure 8 shows how this generic service framework operates on an activity level. A user or client triggers a standard web request via HTTP on the feedback service. The specific DSL programmes are triggered by a simple mapping of the request. Parameters provided are transformed as needed and supplied to the DSL programme. Then, the DSL programme is executed and results will again be transferred to the client.

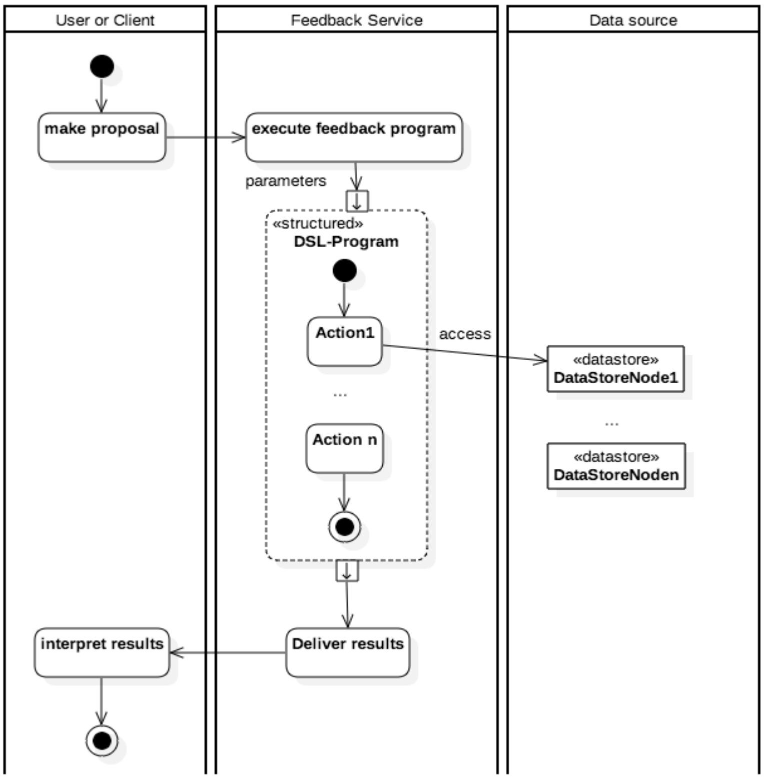


Figure 8. Generic operation of Feedback Service framework.

The feedback service is implemented as a micro-service embedded in the smarticipate platform environment. Figure 9 shows a detailed architecture of the feedback service. The feedback calculation process will be triggered by the web application on software clients run by users on desktop computers or smartphones. The application loads data as required from a web server, for example 3D assets such as houses or terrain data. This data is then combined with aerial imagery from a Web Map Service (WMS) service run by the city. When users interact with the app, for example to place new objects in the 3D scene, the feedback service is triggered by the application to perform reasoning and compute feedback for the new object placed. The feedback service then runs a DSL program that may, for example, gather data from a Web Feature Service (WFS) service of the city and returns results as text or 3D geometry.

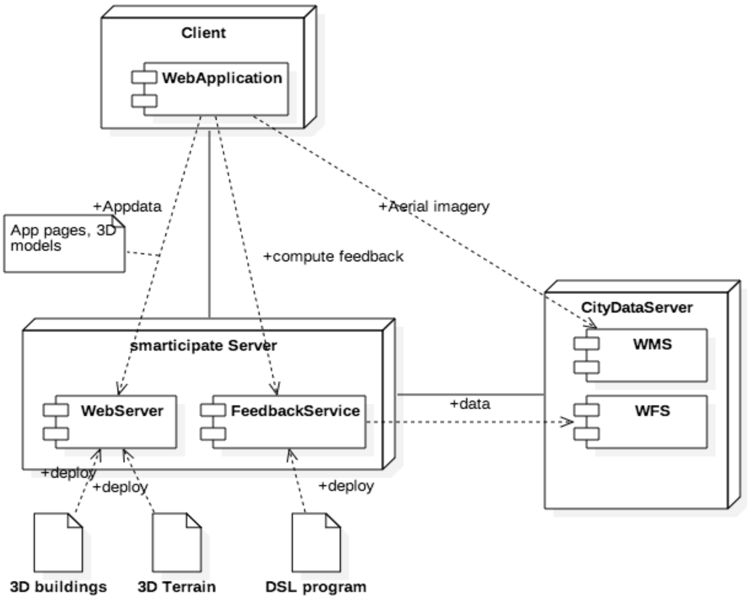


Figure 9. Detailed Feedback service design and dependent environment.

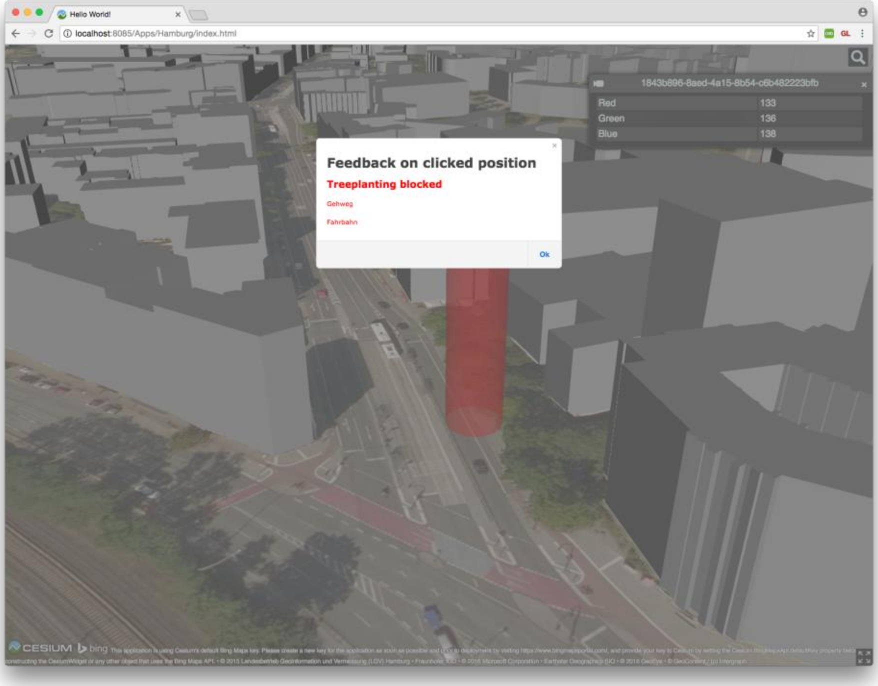
4.2. *Experimental Setup and Proof of Concept of the Feedback Service*

As proof of concept, we concentrate on a simple example of 3D visualisation and feedback service for the above selected use case. It was developed early to demonstrate this feature at various feedback workshops with the objective to (i) determine whether people could understand the concept; (ii) derive domain knowledge; and (iii) find useful examples for a more detailed implementation. So, the proof of concept reveals the extent to which it is possible to implement user needs in feedback feature/service. This provides useful insights for the development of a fully functional system. However, full implementation and evaluation of the feedback service is beyond the scope of this paper.

4.3. *Data for CO_2 Neutral Hamburg–Tree Plantation Use Case Context*

The city of Hamburg provides open data (http://transparenz.hamburg.de/open-data/, last accessed 28 February 2017), which we used to develop an interactive 3D application. This data included (i) the city model of buildings in CityGML; (ii) terrain data as results of airborne laser scans as point clouds; and (iii) aerial imagery directly provided by a WMS server of the city open data portal. The data was processed and transformed to a web-enabled format with tools partially developed for this project. The client is a web application running in a web browser, which uses special formats such as GLTF for 3D assets or quantised-mesh as terrain data. Also, sophisticated data streaming techniques are used to ensure a good user experience. For proof of concept, currently only a block-oriented building model at Level of Detail 1 (LOD1) is available. However, this 3D layer can be replaced with a more detailed model including detailed roofs and façade elements i.e., LOD2, if available.

The Web Browser is the client run-time environment, which loads the application and data to display from several services. The Building Service streams the Hamburg 3D model to the cesiumjs (https://cesiumjs.org, last accessed: 27 February 2017) based web client. The Terrain Service streams terrain data to the client and the client maps aerial imagery from the Hamburg WMS service to the terrain. When a user clicks on a position in the 3D map to check if planting a tree is possible, the position on the map is sent to the Feedback Service to check if there are obstacles (or constraints as defined in DSL) nearby. The data in this case is loaded from a Geoserver (http://geoserver.org, last accessed: 27 February 2017) serving Hamburg street cadastre data. The results of the check are provided to the client along with an explanation why planting a tree is not advisable. The result is, for now, displayed by a simple dialogue and a red coloured surrogate on the clicked position as shown in Figure 10.



1843b896-8aed-4a15-8b54-c6b482223bfb

Red	133
Green	136
Blue	138

Feedback on clicked position

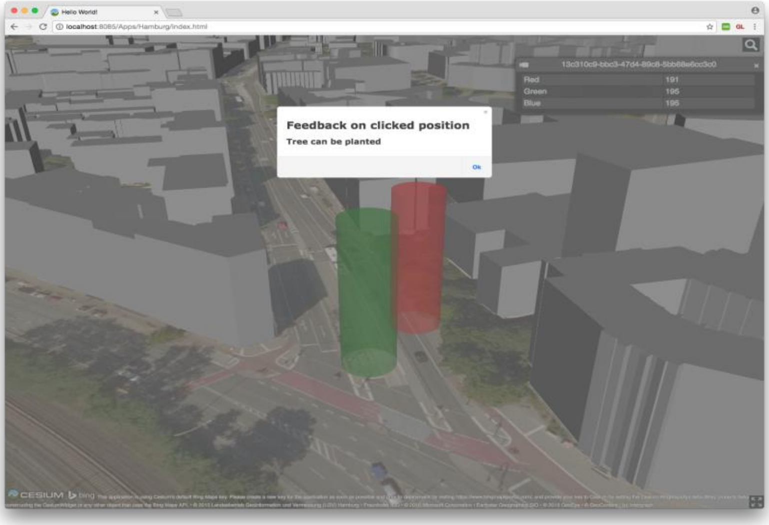
Treeplanting blocked

Gehweg

Fahrbahn

Figure 10. Negative Feedback on a selected position.

In this proof of concept, we defined that objects such as roads, pedestrian ways, crossings and bicycle lanes are unsuitable locations, but we deliberately excluded bus lanes for demonstration. While Figure 10 shows a negative result (i.e., tree plantation is not possible due to freeway (Fahrbahn-German/local language) or pedestrian way (Gehweg-German/local language)), Figure 11 shows a positive result when clicking on a bus lane with a green surrogate object on the clicked position. Please note that the above example represents a very simple scenario. However, a full version of the feedback service intends to cover more detailed feedback, including suggestions of alternative places for tree plantation, cost and budget information, etc. Also, this feedback does not mean that a final planning decision has been made. Rather, this feedback is an early suggestion which can help to inform citizens as to why their proposals are not feasible and what alternatives are available to make the proposal feasible. This also means that once citizens are aware of the limitations in their proposals, they can rectify them and then prepare and submit a formal planning application to their local city administration. This formal application can also include all the participatory evidence collected through the smarticipate platform that will enable city administration to make the final decision.



13c310c9-bbc3-47d4-89c8-5bb68e6cc3c0

Red	191
Green	195
Blue	195

Feedback on clicked position

Tree can be planted

Ok

Figure 11. Positive Feedback on a selected position.

5. Discussion and Lessons Learned

The smarticipate platform has an ambitious list of features as depicted in Figure 6. This indicates the significance of the smarticipate platform in bridging the gap between top-down and bottom-up citizen participatory practices and promotes open governance. These features respond to a detailed list of requirements defined by smarticipate case study city administrations and their citizens. For citizens, there will be a list of topics, proposals, and various features including adding new data, changing existing proposals, or generating alternative proposals, tracking progress of a proposal, sharing their ideas with other user groups, etc., (see Figure 6). Not all these features are straightforward to design and implement and their realisation requires careful research and analysis of the problem domain. For instance, the proof of concept of the automated feedback feature is elaborated with the objective to highlight its benefits and complexity involved in developing such a feature for various domain topics. This is mainly because generating alternative proposals with knowledge-based feedback using visual 3D spatial models requires rich open datasets, domain knowledge, rules and high performance computing infrastructure to process queries of many users. In the following section, we critically assess the strengths and benefits of the smarticipate platform. In addition, we elaborate challenges based on the experience of designing the smarticipate platform and developing the proof of concept of the automated feedback feature.

The above sections demonstrate that obtaining interactive and automated feedback using visual 3D spatial proposals enhances the ability of participants to gain required information immediately to enable them to make informed decisions. We consider that, at later stage, this automated feedback can be 'knowledge-based feedback' due to its capability to process data by applying domain rules and then generate information for the end users. These rules and generated information can be preserved for additional analytics e.g., identifying similar proposals or topics and sharing the results with appropriate stakeholders will improve the efficiency of the system. The 3D modelling enhances the ability of the end users to visualise the proposals and have better understanding of what is being proposed. During feedback workshops, users appreciated the use of 3D spatial models as it helps to provide spatial context to a proposal. It also enhances the overall understanding of the impact of the proposal in the urban neighbourhood. The automated feedback feature analyses the feasibility of the proposal by comparing it against pre-defined rules and suggests reasoning. This helps citizens to make informed choices for a specific proposal without needing the expert's help. This, however, mandates that sufficient domain-specific languages with domain vocabulary and knowledge needs to be provided to handle a specific topic and its associated proposals.

Sharing these proposals with other user groups raises awareness among the residents and enables others to provide their inputs to shape or transform the proposal as appropriate. The sharing of topics and proposals with selected user groups provides the opportunity to initiate consultation with residents who are directly affected by the proposal or to gain suggestions from the wider community. Such an approach complements traditional consultation meetings where community representativeness and a higher participation rate have always been an issue. This means that making these proposals accessible through web or smartphones provides flexibility to citizens to give their opinion about a planning proposal. This promotes a bottom-up open governance that enables citizens to co-create, co-design and participate in planning processes.

From a technical perspective, the use of DSL enables domain experts to define domain vocabulary and rules in a high order descriptive language that provides higher level abstraction. This approach helps in capturing domain-specific information only and hides irrelevant details. This enables users to leverage sophisticated technologies in a transparent way, for example rule-based systems or geographical information systems or 3D applications. At the moment, DSL can be defined by an IT expert or domain expert but not citizens. As future work, other approaches will be investigated to enable citizens to effectively use DSL.

The micro-services-based architecture allows the smarticipate platform to be extended with new features in a more convenient way, as integration of third party services does not require changes in already deployed software. This approach is also used to integrate current legacy IT systems of cities into the smarticipate platform. For example, a highly-ranked proposal, with all participatory evidence, should be submittable to a city's planning application system.

The use of HTML5 for the Progressive Web App development model and Single-Page-Application design using various libraries e.g., leaflet, React, etc., facilitate the development of interactive, responsive, reliable and attractive smarticipate applications. This means that these applications can be rendered on different screen sizes i.e., smartphones, tablets or PC. Further, the docker-container-based approach promotes high modularization of the code base and reusability of existing components and deployment of smarticipate applications in different settings. The aim is to produce training videos so that all users should be able to learn and use the system. This will be part of the future work and these videos will be available when frontend and backend services are fully ready for production.

Based on the above Experience, below We Discuss Some Expected Challenges as Future Research Directions

Co-creating urban design proposals and performing automated analysis for knowledge generation as feedback is not straightforward. The above proof of concept demonstrates the tree plantation use case with well-defined rules and domain vocabulary. However, it is observed in the other use cases of the smarticipate project e.g., fully-open domain-independent proposals for building retrofitting, that there are certain limitations and challenges which may affect the full-scale implementation of required capabilities in the automated feedback feature. This is mainly due to the following factors:

Validation of Domain Knowledge as Vocabulary and Rules

Automated feedback can be generated by using domain-specific vocabulary and applying application rules but these rules require domain knowledge. This domain knowledge can be acquired from domain experts, policy documents or scientific publications. However, correctness and completeness of such vocabulary and rules require validation by domain experts, which is difficult to achieve due to their limited availability.

Tools for DSL Engineering

Defining a suitable DSL and finding associated tools is also necessary. These tools should handle the complexity of defining rules and vocabulary, which can be used by the smarticipate platform to perform reasoning, analytics and feedback generation.

Availability and Accuracy of Spatio-Temporal Data Elements and Deriving Tacit Knowledge

Open government data portals are gradually increasing the variety of datasets and are also producing more recent data (in some cases real-time data, e.g., available parking places in public car parks) available for use by citizens or other business organisations. However, not all data is geo-tagged (with accuracy of centimetres) and many datasets are not updated regularly. This makes it challenging to correlate different datasets for specific geo-coordinates and verify the accuracy and suitability to generate approximate feedback output. Also, without semantic representation or Linked Open Data [30], it can be challenging to accurately use such data for generating analytics and knowledge-based feedback. For example, data may represent streets as exact geometry or polyline, and exact positions of traffic-lights (down to cm) or exact coordinates for the centre of a road crossing may not be available. The impact of the absence of such rich details may result in the generation of erroneous or approximate automated feedback. For such datasets, uncertainty must be considered by putting buffers around such elements and needs to be made transparent in the results, e.g., a fuzzy factor could be introduced. This suggests the need of pre-processing datasets by using a toolkit to integrate and transform data into a useable, highly detailed format.

Visualisation of 3D Models on Different Platforms and Screen Sizes

The capability to visualise urban proposals using 3D models greatly improves the ability of citizens to understand the context of the proposal. The impact on the planning and the results leading to an assessment should be more visual. This means that the influences should also be made visual, for example, by colouring them and not only showing a surrogate such as a coloured cylinder as shown in Figures 10 and 11. With 4 G and 5 G mobile networks and smartphones and tablets having entered the consumer market, it is essential that the smarticipate platform has responsive design and can be used on smartphones, tablets or PCs. This requires careful User-Experience (UX) design principles for the User Interface to accommodate complex 3D models and alternative scenarios building on different screen sizes. The smarticipate platform already handles this by designing wireframes and performing controlled usability studies.

High Performance Computing for Immediate Knowledge Generation and Feedback

End users expect knowledge-based feedback on their proposals immediately, e.g., within a few seconds. This would require high performance computing (or GPU) to process multiple data variables, rules and 3D visualisation to generate knowledge-based analytics as feedback.

Granularity of Data and Rules

The use case-based approach adopted by the smarticipate project is useful to first test the challenges and limits of such a system on a selected domain (e.g., trees, or buildings) with pre-defined rules. This will provide useful insights into the strengths and limitations of applying such a system on other domains. Also, whilst there are pre-defined technical rules used by the platform, the possibility for domain experts to write new rules in high-order DSL based on the policy of the city is also part of the architectural design so that existing rules can be updated and new rules can be defined. Also, domain-specific rules are defined at different levels: (i) *fully-automated* where all required data is available and rules are fully specified to generate automated feedback without expert advice; (ii) *semi-automated* where partial data or rules exist and some expert advice is needed to generate feedback; (iii) *manual* where most data is either missing or is not in the required format and no proper rules are defined will be dependent on the expert advice for feedback generation.

In general, the feedback service must be very clear regarding the quality of the results provided. It is not very likely that, in any possible case, all details are given in such a way that the feedback is deterministic and error-free and this must be conveyed to observers. Missing aspects or missing data can lead to questionable results and to counter this, the system needs to provide, at any time, an explanation regarding which data was used and which rules have led to the result. For example, in the tree plantation use case, there may be circumstances beyond control, which hinder the planting of a new tree, which were not known due to a bad data situation, e.g., unknown utility pipes or hazardous ground below the area. It must be clear that a service works on models which make assumptions and those assumptions must be transparent when interpreting the results.

6. Conclusions and Future Work

The smarticipate platform has the potential to transform city governance by facilitating both top-down and bottom-up decision making. It provides a citizen participation and communication platform to create new proposals using open data for a given topic by using smartphones, tablets or PC. The platform architecture is based on a micro-services-based approach that allows the development of individual features as web services with less dependency on each other. Further, the docker-component-based approach allows the deployment of the platform in different settings. This also helps in reusing and extending existing features based on the evolving requirements of cities. It uses Domain-Specific Languages and associated domain rules to analyse a user-defined proposal and generates knowledge about positives and negatives of the proposal as feedback that enables citizens to make informed choices. Further, it allows these proposals to be shared with other selected user groups including citizens and public administrations with the objective to exchange ideas in transforming their local neighbourhoods.

The proof of concept for the automated feedback feature of the selected Hamburg use case demonstrates visual interaction with the platform. This allows the creation of proposals in 3D models that provides detailed context and enhances understanding by other user groups to comment or generate alternative proposals. Though this demonstrates a very basic scenario, it shows that such an approach is effective and can be extended to handle complex scenarios such as building refurbishment, brownfield regeneration, etc. However, the critical analysis in previous sections highlights a number of challenges, which are mainly associated to the quality of open data, availability of 3D data or urban furniture, domain knowledge in DSL, domain rules, etc. Our future work is to investigate the feasibility of the feedback feature on other topics, such as the Rome use case of building refurbishments.

Acknowledgments: The research work presented in this paper is partly funded by the European Commission's Framework Programme H2020 Smarticipate project. Grant # 693729. Authors also acknowledge the support of consortium partners who participated at different stages of the project.

Author Contributions: Zaheer Khan initiated and provided conceptual basis for the article after collecting smarticipate project requirements and then writing abstract, introduction, related work, smarticipate methodology, discussion and conclusions. Jens Dambruch contributed to Domain-specific Languages, gathered additional requirements and examples in workshops for this topic, provided the technical proof-of-concept and solution design. Zaheer Khan and Kamran Soomro were responsible for requirements management in the Redmine requirements system. Peter Fröhlich and Andreas Sackl are responsible for designing front end of the smarticipate platform. All contributed to section 3 about the system concept and technologies. Jan Peters-Anders and Andreas Sackl wrote about the system architecture. Anton Strasser and Simon Templer provided inputs about the technology used to implement the platform. All provided inputs to discussion, conclusions and further work.

Conflicts of Interest: Authors declare that they do not have any conflict of interest.

References

1. Arnstein, S. A ladder of citizen participation. *J. Am. Inst. Plan.* **1969**, *35*, 216–224. [CrossRef]
2. Khan, Z.; Ludlow, D.; Loibl, W.; Soomro, K. ICT enabled participatory urban planning and policy development: The UrbanAPI project. *Trans. Gov. People Process Policy* **2014**, *8*, 205–229. [CrossRef]
3. Dambruch, J.; Krämer, M. Leveraging public participation in urban planning with 3D web technology. In Proceedings of the 19th International ACM Conference on 3D Web Technologies, Vancouver, BC, Canada, 8–10 August 2014; ACM: New York, NY, USA, 2014; pp. 117–124.
4. Semanjski, I.; Bellens, R.; Gautama, S.; Witlox, F. Integrating Big Data into a Sustainable Mobility Policy 2.0 Planning Support System. *Sustainability* **2016**, *8*, 1142. [CrossRef]
5. Open Government Data. Available online: https://opengovernmentdata.org/ (accessed on 23 February 2017).

6. Khan, Z.; Liaquat Kiani, S.; Soomro, K. A framework for cloud-based context-aware information services for citizens in smart cities. *J. Cloud Comput. Adv. Syst. Appl.* **2014**, *3*, 14. [CrossRef]
7. Smarticipate Project. *Smart Open Data Services and Impact Assessment for Open Governance*; European Commission H2020 Programme. Available online: https://www.smarticipate.eu/ (accessed on 24 February 2017).
8. Soomro, K.; Khan, Z.; Ludlow, D. Participatory governance in smart cities: The urbanAPI case study. *Int. J. Serv. Technol. Manag.* **2017**. (in press).
9. Berntzen, L.; Johannessen, M.R. The Role of Citizen Participation in Municipal Smart City Projects: Lessons Learned from Norway. In *Smarter as the New Urban Agenda—A Comprehensive View of the 21st Century City*; Public Administration and Information Technology; Springer: Berlin, Germany, 2016; Volume 11, pp. 299–314.
10. European Commission. *A Vision for Public Services*; DG-CNECT: Brussels, Belgium, 2013.
11. Misuraca, G.; Broster, D.; Centeno, C.; Punie, Y.; Lampathaki, F.; Charalabidis, Y.; Askounis, D.; Osimo, D.; Szkuta, K.; Bicking, M. *Envisioning Digital Europe 2030: Scenarios for ICT in Future Provenance and Policy Modelling*; Misuraca, G., Lusoli, W., Eds.; European Commission: Brussels, Belgium, 2010.
12. Insight Project. Policy Modelling and Governance Tools for Sustainable Urban Development—State of the Art and Future Challenges. Available online: http://www.nommon-files.es/working_papers/INSIGHT_PositionPaper_Oct2013.pdf (accessed on 8 February 2017).
13. Bunt, L.; Harris, M. Mass Localism: A Way to Help Small Communities Solve Big Social Challenges: Discussion Paper. Available online: http://www.nesta.org.uk/sites/default/files/mass_localism.pdf (accessed on 8 February 2017).
14. European Commission. *Powering European Public Sector Innovation: Towards a New Architecture: Report of the Expert Group on Public Sector Innovation*; European Commission: Brussels, Belgium, 2013.
15. Berman, T. Public participation as an instrument for incorporating local knowledge into planning processes. In Proceedings of the State of Australian Cities Conference, Queensland, Australia, 9–11 December 2015.
16. Beebeejaun, Y. *The Participatory City*; Jovis Verlag GmbH: Berlin, Germany, 2016.
17. Ruppert, T.; Dambruch, J.; Krämer, M.; Kohlhammer, J.; Balke, T.; Gavanelli, M.; Bragaglia, S.; Chesani, F.; Milano, M. Visual Decision Support for Policy Making: Advancing Policy Analysis with Visualization. In *Public Administration and Information Technology*; Janssen, M., Ed.; International Publishers Association: Geneva, Switzerland, 2015.
18. Kramer, M.; Khan, Z.; Ludlow, D. Domain-specific languages for agile urban policy modelling. In Proceedings of the 27th European Conference on Modelling and Simulation (ECMS), Alesund, Norway, 27–30 May 2013; pp. 673–680.
19. Krämer, M. Controlling the processing of smart city data in the cloud with domain-specific languages. In Proceedings of the IEEE/ACM 7th International Conference on Utility and Cloud Computing, Washington, DC, USA, 8–11 December 2014; pp. 824–829.
20. Future of Planning. *Future of Planning—State of the Art Innovations in Digital Planning*; Future City Catapult: London, UK, November 2016. Available online: http://futurecities.catapult.org.uk/2016/12/01/future-planning-state-art-innovations-digital-planning/ (accessed on 25 January 2017).
21. Cohen, B.; Almirall, E.; Chesbrough, H. The City as a Lab: Open Innovation meets the Collaborative Economy. *Calif. Manag. Rev.* **2016**, *59*, 5–13. [CrossRef]
22. Khan, Z.; Ludlow, D.; Loibl, W. Applying the CoReS requirements development method for building IT tools for urban management systems: The UrbanAPI project. *Theor. Empir. Res. Urban Manag.* **2013**, *8*, 25–59.
23. Redmine. Redmine—Project Management System. Available online: http://www.redmine.org/ (accessed on 8 February 2017).
24. Schwaber, K. SCRUM Development Process. In Proceedings of the Workshop on Business Object Design and Implementation at the 10th Annual conference on Object-Oriented Programming Systems, Languages and Applications (OOPSLA'95), Austin, TX, USA, 15–19 October 1995.
25. Khan, Z.; Ludlow, D.; Rix, J. Applying the criteria indicators and metrics evaluation method on ICT research: The HUMBOLDT project. *Res. Eval. J.* **2012**, *22*, 30–40. [CrossRef]
26. Khan, Z.; Ludlow, D.; Caceres, S. Evaluating a collaborative IT based research and development project. *Eval. Progr. Plan.* **2013**, *40*, 27–41. [CrossRef] [PubMed]
27. Namiot, D.; Sneps-Sneppe, M. On Micro-services Architecture. *Int. J. Open Inf. Technol.* **2014**, *2*, 24–27.

28. Malewski, C.; Dambruch, J.; Krämer, M. Towards Interactive Geodata Analysis through a Combination of Domain-Specific Languages and 3D Geo Applications in a Web Portal Environment. In Proceedings of the REAL CORP Conference, Ghent, Belgium, 5–7 May 2015; pp. 609–616.
29. Nicola, A.D.; Missikoff, M.; Navigli, R. A software engineering approach to ontology building. *Inf. Syst.* **2009**, *34*, 258–275. [CrossRef]
30. Linked Open Data. Linked Open Data: The Essentials—A Quick Start Guide for Decision Makers. Available online: http://www.semantic-web.at/LOD-TheEssentials.pdf (accessed on 28 February 2017).