# Web-Augmentation, Design Thinking, and Collaboration Engineering to Foster Innovation in DSS for Agriculture: A Case Study

Matías Urbieta[1,2], Sergio Firmenich[1,2],  Pascale Zaraté[4], Alejandro Fernandez[1,3]

[1]LIFIA, Facultad de Informática, Universidad Nacional de La Plata
[2]CONICET, Argentina
[3]CICPBA, Argentina
```
[matias.urbieta, sergio.firmenich, alejandro.fernandez]
                @lifia.info.unlp.edu.ar
```
[4]Toulouse University, IRIT, 2 rue du Doyen Gabriel Marty, 31043 Toulouse Cedex 9 , France,
```
pascale.zarate@irit.fr
```

**Abstract.** Decision making in agriculture increasingly relies on software, for example to gather important information or to weight alternatives. Information systems innovation in agriculture is a challenging and very active area. Existing software products, frequently implemented as web applications, are found to lack functionality, for example to support collaboration. Augmenting the web is a widely adopted technique for enhancing existing applications with new features which are not available out-of-the-shelf. Design thinking has proven to be an effective tool to support innovation on many domains. Collaboration Engineering is an effective means to reuse design experience of collaboration strategies. This work presents an approach to involve end-users in enhancing exiting web software to produce incremental innovations. The approach relies on Design Thinking, Web Augmentation and Collaboration Engineering. The approach was successfully tried in a case study letting end-users add collaboration support to a system that did not provide it.

**Keywords:** Agriculture, software, augmentation, collaboration.

## 1    Introduction

Information technologies offer a rich variety of tools to help practitioners make decisions. These tools, generally called Decision Support Systems (DSS) [1], [2], are available in multiple domains. In current times, these tools take the form of web-applications. Although some decisions a person makes may appear unconscious (such as those made while driving), other decisions require that the person stops, reflects, and selects what is considered the most effective alternative. The strategy a person uses depends on the importance of the decision. Intuitively, the more important the decision, the harder the decision maker needs to think about it. A complex decision requires that one applies specific knowledge, incorporates multiple perspectives, considers various

sources of information, and carefully weights alternative courses of action. Given its complexity, this process often requires software support. Moreover, when decisions involve multiple stakeholders (as most decisions taken in the context of work) collaboration support becomes necessary. However, collaboration isn't always adequately supported.

In agriculture, decisions such as campaign planning, soil management, handling crop health, selecting crop varieties, and commercialization were once made on the basis of traditional practices. However, globalization and constant climate change have become a source of complexity, uncertainty, and risk that challenge traditional methods.

Farm Information Management Systems (FIMS) are software systems that support decision making in farms. As reported in a recent review [3], FIMS have evolved from the original simple record-keeping programs into sophisticated and complex systems to support production management. Current FMIS adopt and adapt features that are common to ERP systems, such as operations management, finance, inventory, reporting, machinery management, human resource management, sales, traceability, and quality assurance. In addition, they offer a set of specific features such as best practice, site management, and precision farming. FIMS were introduced in the 70's [4], and nowadays they are widely available. Nevertheless, FIMS adoption is still limited. In Brazil, for example, contract adjustments and farmers' experience were found to have a negative impact on adoption of FIMS [5]. A study that surveyed ICT adoption in New Zealand and Uruguay [6] proposed a model to understand adoption. The model states that farmer's attributes such as objectives, personality, education, learning style, and skills directly influence adoption. Therefore, the authors conclude that software developers must work with farmers; both during design, and later providing training and support. Moreover, they argue that systems need to configurable to suit a range of farmer characteristics.

Information is central to decision making [7]. Besides information available locally, which might be recorded with support of a FIMS, agriculture decision making requires timely information that is normally available elsewhere. The work presented in this article places emphasis in information available on the World Wide Web, as it is one of the most used sources of Big Data that is not always considered in the kind of systems described above. The Web is a constantly growing source of information, created by collaborating individuals around the world. However, most of the information published on the Web is isolated in silos, scattered across multiple sources, or unfit for use [8]. Moreover, as practice knowledge is naturally distributed among stakeholders, collaboration becomes paramount. During campaign planning, for example, farmers must evaluate different sort of seeds that can be used in the plot. They must consider the production history of the plot, the fruits or vegetables being produced in nearby plots, and the potential for the occurrence of plagues. This information gathering task requires to browse different websites where information about seeds is captured including vendor information, community reviews and success cases. To make matters more complicated, the activity if often performed collaboratively by farmers, agricultural engineers, and procurement agents without any collaboration tools.

In the context of the RUC-APS project [9], researchers have met with farmers in order to gather requirements for tools to support collaborative decision making based on information available on the web. In order to involve stakeholders from the start of the analysis and design phases, a design thinking approach was followed. Design thinking [10] is a set of strategies commonly used by designers, especially in the presence of wicked problems. Design thinking consists largely of three interrelated processed. Inspiration is the process that brings together designers and users. Ideation is the process in which designers and users collaborate to imagine, prototype and test solutions to key problems identified during the inspiration process. Solutions that pass the tests, are then implemented into products as part in the context of the Implementation process.

This article presents a case study that shows how design thinking, collaboration engineering patterns, and Model-Driven Web Augmentation may be combined to produce a Minimal Viable Product (MVP) for collaborative campaign planning with information from the web. The rest of this document is organized as follows: Section 2 provides background. Then, section 3 presents an overview of the approach. Section 4 shows a comprehensive example. And Section 5 concludes and talks about the future works.

## 2 Background

This section provides context to the information discussed throughout the research paper. The Background includes a brief description of main building blocks of the approach followed.

### 2.1 Collaboration Engineering and Patterns

Systems designers have learned the value of documenting, in the form of design patterns, proven solutions to recurring problems [18]. Design patterns increase productivity, bring quality, and support effective communication among designers and developers. Patterns exist to support and speedup the design of computer mediated collaboration [19]. Researchers observed that, in effective collaborative processes, participants engage in recurrent patterns of collaboration [20]. There is a "diverge" pattern of collaboration characterized an increase in the number of available choices (e.g., ideas). It follows a "converge" pattern of collaboration where participant engage to obtain less choices of higher value. These choices are then "organized", as participants get a better idea about the relationships among choices. The resulting choices are "evaluated" to better understand how each of them contributes to the goal. To complete the cycle, participants engage to "build consensus".

Researchers also observed that the success of these collaboration initiatives was strongly related to the availability of a facilitator whose role is to effectively guide participants though the process. On the basis of this understanding, they proposed Collaboration Engineering [20][21][22]. Collaboration engineering provides a series of strategies and tools that capture effective collaboration strategies, in a way that makes results replicable. A key element in collaboration engineering is the Thinklet. A

ThinkLet is packaged facilitation intervention that creates a predictable, repeatable pattern of collaboration among people working together toward a goal [22]. Thinklets consist of three components: a) a tool (e.g. a software); b) a configuration for the tool; c) a facilitation script. The creators of this methodology published a catalog including detailed description of 40 ThinkLets. However, the methodology allows for new ThinkLets to be discovered and recorded.

## 2.2 Design Thinking

Design thinking [10] is a set of strategies commonly used by designers, especially in the presence of wicked problems. It is widely used to foster innovation. Although there are various models to describe the design thinking process, it largely consists of three interrelated processes or activities: Inspiration, Ideation, and Implementation. As a general rule, these activities are seen as a phase of divergent thought followed by a phase of convergent thought.

Inspiration is the activity that brings together designers and users. The goal of inspiration is to identify, understand, and record key challenges or problems. Empathy and collaborative, multidisciplinary work are key elements during this activity. Multiple tools and strategies. can be used to support this activity, for example Journey Mapping.

Ideation is the process in which designers and users collaborate to imagine, prototype and test solutions to key problems identified during the inspiration process. Many proposals will be produced, only a few of which will have a change of being discussed with the end customer. Designers must turn to prototyping techniques that require low effort (for example paper mockups). However, there is a trade-off between low effort, and the ability of the "low-fidelity" prototypes to convey the solution idea with enough clarity. For the case of software solutions, the ability to interact with the prototype can make a big difference. Model-driven web augmentation, and collaboration patterns are tools that can support the ideation activity.

The prototypes that pass the ideation phase have a chance of becoming final products. During the implementation activity, stakeholders decide how much effort they will invest to develop the selected ideas. As they still don't know how well they will perform, they may decide to start small and test, for example, in pilot scale. During implementation, model driven web-augmentation can also play an important role.

## 2.3 Group Decision Support Systems

Increasingly challenging global and environmental requirements have resulted in agricultural systems coming under increasing pressure to enhance their resilience capabilities. As stated by Hernandez et al. colleagues [16], integrated solutions are necessary to support knowledge-management, collaboration, risk management and regulation management across agriculture stakeholders. Such solutions need to consider both the capabilities of the web as a source of reusable information, and the web as platform on which systems (applications) operate. As agricultural production systems are rich networks of independent actors, collaboration becomes paramount.

Group Decision Support Systems are software systems designed to help multiple stakeholders during decision making processes. They offer support for communication, collaboration, coordination, information retrieval and, in some cases, incorporate advances in operational research regarding the solution of multi-variable problems. Different GDSS emphasize different perspectives of the challenge. Some of them accentuate the importance of problem modeling, some of them focus on facilitating collaboration among stakeholders and some others adopt a MultiCriteria approach [17].

## 2.4    Web Augmentation

Since almost twenty years, hypermedia and web communities are dealing with the alteration and integration of existing third-party applications, which was coined as Web augmentation [11], [12], and currently this alteration is mostly performed be means of Web browser extensions. When Web browsers reached widespread use and social networks started to arise, also the first actual communities around software for adapting existing Web sites emerged. For instance, an important community emerged around *userscrtips*, which are JavaScript script that are installed and executed on the browser when a Web page loads, allowing its alteration.

Nowadays, the techniques behind Web augmentation are used by a vast number of users that may choose among thousands of extensions for adapting Web content may be found at the Web browser stores, and significant developer communities support some of these tools. For instance, the Userstyles community (http://userstyles.org) offers a wide number of scripts that augment Web sites by adding further CSS specifications that change the content presentation. Userscripts communities, such as Greasyfork (https://greasyfork.org/), offer repositories of scripts with a wider spectrum of purposes, since they support different weavers of JavaScript code (e.g. GreaseMonkey or TamperMonkey); therefore, it is possible to change not just the style but the content and behavior of a Web page.

End-users and other stakeholders with programming skills may interact in such communities for the creation, sharing, and improvement of specific augmentation artifacts [13]. In all these communities, no matter which tool they support, there is a dependency between users with and without programming skills, since not all of them can implement the solutions they need and ask others for help. In this light, some research works proposed End-User Development (EUD) approaches to let users specify their own augmentation artifacts; these works are discussed in the related work section [14], [15].

Although the existence of EUP environments makes the alteration of UI straightforward, complex requirements such as collaboration, require more than just stand-alone client-side scripts [11]. In this sense, involving a method to integrate back-end counterparts (i.e., a web application working on a server) is needed. Still with the focus on fast prototyping, it is well-known that MDD approaches are more productive, therefore, this work proposes a Model Driven Web Augmentation approach, which is based on the models at both sides, front-end and back-end.

## 3    The approach in a nutshell

Existing software engineering practices stablish the identification of system requirements as the starting point. Analysts record what the prospective users expect the system to do on their behalf. This practice assumes that the prospective users have a concrete list of requirements that need to be fulfilled, and that they understand what a software system can do for them. Agriculture, however, is a domain in which penetration of modern Web technologies is low. Apart from early adopters and experimental programs, most farmers are not aware of the potential that information technologies can offer. In this context, traditional requirements elicitation techniques do not apply, and systems design becomes an exploratory and iterative innovation process that calls for adequate tools. Nevertheless, if a set of Web applications related to the desired domain is available, they could be used as a basis to speed up information requirements gathering. In this way, by augmenting these Web applications new requirements may be considered and modeled.

Previous work [23] introduced the foundations of an approach for modeling complex augmentation that comprises both client and server-side artefacts. The approach takes advantages of model-driven claimed benefits, such as high-productivity and less error-prone than traditional development approaches that start from scratch.

This work, presents an approach giving support to complex business process comprising inter-site information and groupware requirement gathered from end-users. It relies on a design thinking approach for extracting and priorizing requirements and uses Model-Driven Web Augmentation technologies for producing an MVP (Minimum Viable Product).

This work is based on the design thinking approach proposed by Hasso-Plattner Institute of Design at Stanford which considers 5 steps: Empathise, Define, Ideate, Prototype, and Test. Figure 1, introduces the main steps of the approach and related artifacts. In the image tools used in the approach as well as players who perform the activities are shown.
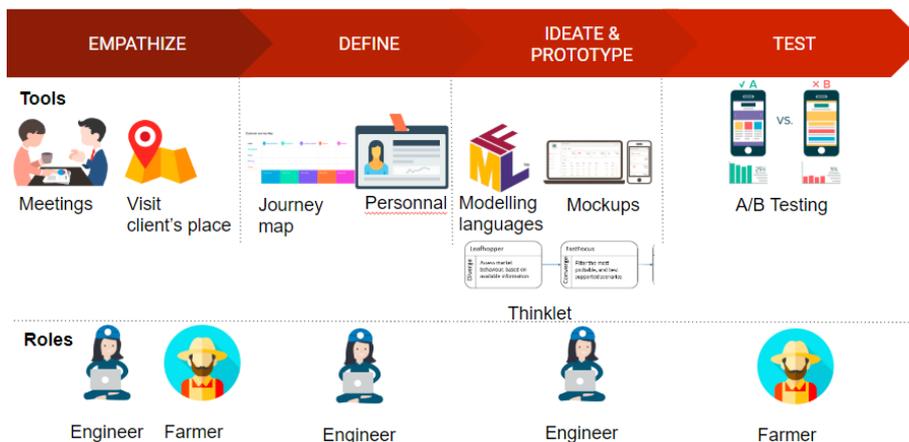


**Fig. 1.** The Design Thinking process based on MDWA

## 3.1    Empathize

The Inspiration step focuses on understanding the end-user situation and identifying the opportunities to improve the business experience. The information gathering requires visiting end-user's place, interview stakeholders, and perform a business activity on behalf of the end-user. In this way, analysts can experience and explore the whole business context.

In order to get the most from the interviews, the analyst does not only focus on the business processes and economic goals of the business, indeed he must maximize his empathy understand psychological and emotional needs of stakeholders and thus understand what is done, why is done in that way and how stakeholders feel about the experience highlighting what is really meaningful.

## 3.2    Define

The problem statement is defined in this step. The team describes the issues that need to be addressed in each activity of the business process. The captured understanding during meetings and visits can be documented using the journey map tool which describes the business experience points, pains, opportunities, etc.

In this stage the co-creative tool Journey Mapping is used for gathering processes information which is developed jointly with stakeholders. One of the requirements is the definition of the personas profile used for depicting the main business actors. According to the experience exposed in this step, the process activities are ranked somehow it is clear if it has a positive or negative contribution to the overall experience. Additionally, the pains are outlined and opportunities are presented. The resulting Journey Mapping document along with personas' profiles are later during the ideation step.

## 3.3    Ideate and prototype

The ideation is a twofold process where firstly the collaboration requirements are analyzed and designed, and secondly these are modeled using a Model-Driven Web Augmentation Approach.

### Collaboration engineering

As people move together towards a common goal (e.g., to make an informed decision) they engage in a process in which they contribute with their knowledge, effort, and resources. This collaborative process can be characterized as consisting of the following phases: understanding the problem, developing alternative solutions, evaluating alternatives, choosing alternatives, making a plan, taking action, and monitoring results.

Once the journey map is elaborated, a throughout analysis of pains is performed in such way collaboration opportunities are identified. By using ThinkLets tool, the software engineers ideate improvements in the existing business process.

## A Model-driven Web Augmentation approach

A Model-Driven Web Augmentation approach is used by Software engineers to develop complex inter-site by designing both client and server-side augmentation components. The resulting integration seeks to accomplish business goals through the implementation of business processes. The approach for designing and developing Web augmentations requires following steps:

- Decouple the augmentation from the core application by introducing a design layer (called Augmentation Layer), which comprises additional conceptual, navigational an interface models apart from the ones of the core application. This layer groups all the new elements that will enhance existing Web sites that are used in the business process being augmented.
- Capture the basic conceptual model by tagging the information presented in the core application pages, in a similar fashion to WOA [24]. The lack of access to the underlying business models, which models business entities and their attributes, of the applications being augmented requires the generation of a second conceptual model; the one inferred by the user perception of the augmented. In this process, data elements in the page are tagged and grouped by an augmentation analyst into an entity definition. In such a way, a simplified conceptual model is obtained. The augmentation analyst is a skilled end-user with advanced knowledge of Web Applications, whose goal is to improve a core application. In further steps, the model instantiation in a particular user session will be used for giving contextual information to the augmentation engine by providing model instances information when triggering the augmentation.
- Augmentation requirements are modeled using Web engineering notations (e.g. use cases or user interaction diagrams) and separately mapped onto the following models using the heuristics defined by the design approach (e.g. [25]). Notice that, the augmentation requirements are not integrated into the core requirements model, leaving their integration to further design activities.
- New behaviors, i.e. those belonging to the Augmentation layer, are modeled as first-class objects in the augmentation conceptual model. Such model defines all the objects and behaviors corresponding to the new requirements. Additionally, it may include the core application conceptual classes perceived by the augmentation analyst, allowing defining relationships between the augmentation business model and the core application. Notice that this strategy can be applied to any object-oriented method, i.e., any method using a UML-like specification approach.
- Nodes and links belonging to the augmentation's navigational model may or may not have links to the core navigational model. The core navigational model is also oblivious to the augmentation's navigational classes, i.e., there are no links or other references from the core to the augmentation layer. This principle can be applied in any Web design approach.
- This approach uses a separate specification for the connection between the core and the augmentation nodes. As will be shown later in the paper, the integration is achieved at run-time as part of a client-side weaving engine. Conversely, in

other model-driven approaches, the integration can be performed during model transformation which is not the case of this approach.

- Optionally, the interfaces corresponding to both the application being augmented and the new content expected to be introduced are designed (and implemented). For such task engineers rely on Mockups which are used to document how User Interfaces should look like by example.

In [23] was presented the usage of Web augmentation to give support to the evaluation of raw material used during the vegetables production.

**Testing**

Design thinking proposes a flexible testing session that allows to change the prototype on-the-fly during user tests if this is required. The MDWA approach presented here provides a very convenient environment for this kind of activity. First, the client-side component allows to redefine the augmentation weaving on-the-flight, which is made visually on the browser. Second, using EUD tools for Web augmentation, the UI may be easily. Third, if the augmentation process requires some adaptation on the back-end component, since it is deployed using a MDD approach and cloud services, it may be also altered during the user test session.

## 4    Running example

The green belt of La Plata city is a farming area of approximately 6.000 hectares that provides fresh vegetables to a large part of the population of the Buenos Aires province, Argentina. Tomato and green leafy vegetables are key product of the area. According to the last available census data [26] there were in 2005 over 1000 farms in this area. This section describes how the approach was applied in a mid-size farm in La Plata city. The experience was sponsored by farm's managers and owners giving access to the facilities and promoting meetings with farm's employees.

### 4.1    Empathize

Multiple meetings with farmers, agronomists, and experts were conducted in the green belt of the La Plata city. In all these meetings a recurring problem was the difficulty to assess the plans of the farming community as a whole and consequently adapt one's own farming plan. In Figures 2 and 3, show pictures of meetings celebrated at farm's facilities and at a research lab; respectively.

**Fig. 2.** The farm's manager describes some challenges of producing Tomatoes.



**Fig. 3.** Software engineers interview to farm's stakeholders

As a result, a farmer that decided to plant 10 hectares of tomato, could learn near the time of the harvest that there was an over production of tomato and the price has substantially dropped. There is no up-to-date information regarding the production in this area on which farmers can base their decisions. There is no central organization that helps farmers plan production. Consequently, when farmers make a decision that depends on the projected production of a given crop, they do this by resorting to intuition and talks with a few colleagues.

### 4.2 Define

In figure 4 the resulting journey map is shown (see [1] for further detail). Most important production steps are analyzed by Software engineers who apply collaboration lens to identify opportunities. As journey maps list a myriad of relevant challenges and improvement opportunities, the software engineer agree with farmers on prioritizing and choosing the need to work on because it can be found opportunities from robots to assist manual works relating to harvesting to cultural works training for employees.

### 4.3 Ideate & Prototype

This step shows how collaboration engineering and MDWA can help produce low effort prototypes that feed-back into the innovation cycle.

---

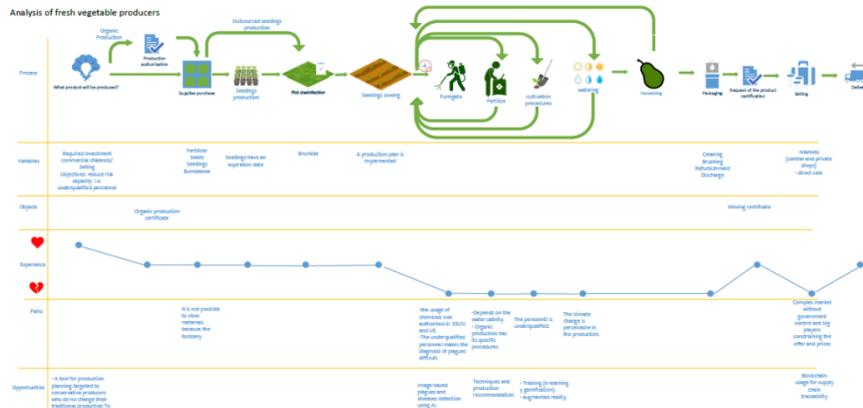[1] Analysis of fresh vegetables producer – a journey map   https://bit.ly/2Ku6Jyl

**Fig. 4.** Journey map of vegetables producers

**Engineering the collaboration between farmers**

Regardless of the higher quality of the tomatoes produced by the farm members, they were not able to obtain a good market price and make a profit. This year they want to plan better and try to predict possible scenarios or outcomes and plan accordingly. They know "everything is on the internet now" so, with the help of a group collaboration engineers they prototype the tools they need to harness the power of the web.
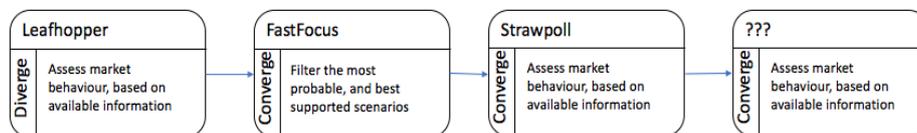


**Fig. 5.** Thinklet chain of collaboration features

Together, they design a collaboration process facilitated by Thinklets implemented on top of web augmentations. The diagram shown in figure 5 using Thinklet Modelling Language provides an overview of the collaboration process they designed.

Next the collaboration scenario is characterized:

**Who**. Members of a community of familiar farmers. They share knowledge to improve everyone's results. They share and exchange their production surplus.

**Objective**. Collaborate during the planning of the upcoming season in order to help each other find the best seeds/seedlings, and other required inputs, and possibly obtain a better price by buying as a group.

**Context**. The group has a Facebook page. They normally use Facebook (posting in the group, messaging) to share information. However, information cannot be organized, ranked, related and turned into plans. "An app to support communities like us would be a total success" they said. They looked for an "app" they can use to plan the season, but nothing seems to be adequate. Therefore, they met with a team of IT guys to create it.

Luckily, they chose a design thinking approach, collaboration engineering patterns, and web-augmentation to build a first prototype. This is what they came up with.

The next section depicts how a model driven approach can speed up the prototype development.


**Augmentation Development**

Previous work [27] introduced the definition of Concern-Sensitive Navigation (CSN) for a Web application when the content, links, and operations exhibited by the core application pages are not fixed for different navigation paths, but instead can change when accessed in the context of various navigation concerns. In CSN, its properties can be slightly adjusted according to the current user's concern; for example, for assisting the vegetable production planning. Web augmentation can be used to implement CSN in Web applications because of the applications' pages are enhanced with new content, navigation, and links according to the task purpose; supporting the underlying user's activity that motivates the navigation. As a result, the user's navigations may help to improve the user's experience while performing the activity.

Considering the scenario described early in this section in the domain of agriculture: planning a vegetable production by a farmer. One of the minor tasks of the planning activity is the research of different products alternatives used during the vegetable production.

As part of the ideation step stakeholders proposed the Far-o-matic tool which is a digital facilitation tool that guides farmers along a multi-step process. The first step was to identify the community needs and the best alternatives to cover them. In the second step, farmers build their plans choosing inputs from the best alternatives already identify. At this point they indicate how much they need and what they will produce. The result is an overview of the potential yields for the whole community and total required inputs. This report is the focus of a shared group discussion, that might drive some farmers to adjust their plans. They might step two and three a couple of times. At the final step, each of them prints a report. The reports include not on-line the plans, but also the shopping cart that can both be used for a common purchase or for individual purchases. The main features of the tool being modelled are:

**Start.** The tool is built on top of Facebook. Whenever farmers log in to Facebook, the tool adds an additional menu entry: Farm-o-matic.
**The wish list**. First, each of them (with a plan in mind) collects products from various web-sites. Products are saved to a shared wish-list. Collected products are visible for everyone. It is possible comment in favor, or against a product. In addition, one can offer an alternative for a product (it can be the same product from a different provider and/or different price, or an alternative product). The result is a list of products, and potential alternatives for some of them.
**Sending the best alternatives to the shopping cart**. Some products in the list have alternatives. Farmers vote, in each case, which alternative they recommend. They can vote only one per group of alternative products. When voting finished, all products are

transferred automatically to the shopping cart. For those products that had alternatives, only the most voted one goes to the cart.

**The purchase order.** Now, farmers express their purchase plans, indicating how much of each product in the shopping cart they need. At all times they can browse and print the global purchase order, and the individual orders.

In order to support abovementioned requirements some design tasks need to be carried out.

## Extracting conceptual model from core application

In the approach presented here, augmentation functionalities might be new behaviors which are added to the conceptual model (and which might encompass many classes). Figure 6 presents the conceptual model composed by three entities:

- Seed to capture alternative species of a given product.
- Pesticide to identify the requirements and constraints of the products and the vegetables.
- Fertilizer to identify the product and used technique.

This model is used for persisting a pre-selection of Seeds, Pesticides and Fertilizers that will be evaluated to define which vegetable will be produced by a farmer.

| Seed | Fertilizer | Pesticide |
|---|---|---|
| oid: integer | oid: integer | oid: integer |
| name: string | name: string | name: string |
| species: string | requirements: string | technique: string |
| price: float | constraints: string | price: float |
| | price: float | |

**Fig. 6.** Conceptual model of products being reviewed by farmers

## Modeling the Augmentation layer

This example introduces an interesting challenge of aggregating information gathered from the navigation of several sites and its persistence. Therefore, the back-end component plays a primary role since it allows to centralize the user's activity information storing the navigation's state defined by navigating several Website.

The Figure 7 presents the navigation model required for supporting the products' information gathering task. The model is based on the IFML standard for describing how the application will be navigated. Each time a user accesses a Web site that contains a Seed, a Pesticide, or a Fertilizer concept, a bucket (Selecting Seed, Selecting Fertilizer, and Selecting Pesticide) of such entity lists the product's details (the concept attributes, such as the name or the price). Besides, it is possible to add the current visited product to the list or delete one product from the list. At any time, the end-user can review current task advance by accessing planning report page through a link present

in each bucket. The flow from Selecting Seed, Selecting Fertilizer, and Selecting Pesticide pages to the Report one allows such navigation.
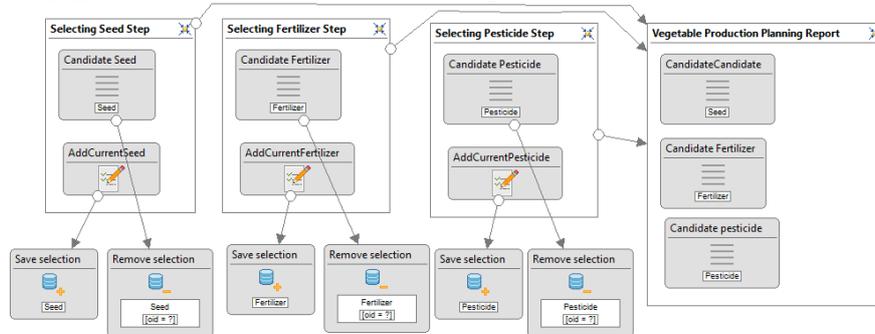


**Fig. 7.** Navigation model depicting how the product selection activity behaves.

The result of enriching a Web site with features that allow the user to gather information, which will be used later, during a complex process that involves several activities in different Web sites is shown in Figure 8. The first step is the Seed selection at Lowe's[2] . Then, several fertilizers are selected by a farmer user to be evaluated picking them from Amazon[3]. As a third step, a support for saving a list of pesticide is added to the TractorSupply[4] web site honoring. The user can access to a summary page where all gathered information is shown and can be used to make a decision for the vegetable to be produced. Note that the navigation must not be performed necessarily in the presented order. Indeed, the user can browse all the applications (back and forth) until he obtains the information he needs for making a decision.

The WebRatio [1] tool supports the IFML approach and the designers can use the tool for modeling the business model and the navigational. The tool generates the Java code related to the modelled entities and relationships as well as the modelled Web application. Moreover, the tool generates the ORM descriptors for Hibernate framework which allows persisting and retrieving objects from/to most of the database engines (i.e. MySQL, Microsoft SQL and Oracle). This approach promotes the use of MDWE approaches for modelling Web Augmentation because the benefits reported on developer's performance. However, the Web augmentation client and server-side artifacts can be developed from the scratch based on software engineering practices.

---

[2] https://www.lowes.com, last accessed July 27th 2017

[3] https://www.amazon.com, last accessed July 27th 2017

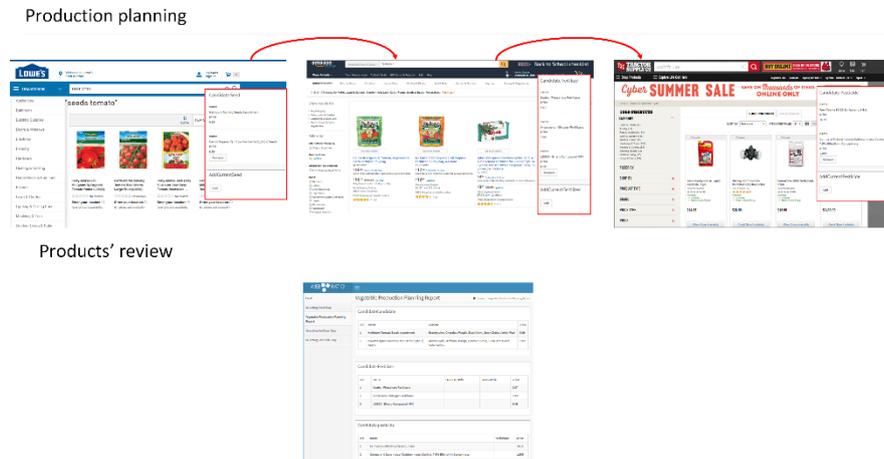[4] http://www.tractorsupply.com, last accessed July 27th 2017

**Fig. 8.** Final augmentation results and gathered information report.

## 5　Conclusions

Decision making in agriculture is as complex as it is in other domain because of the variables involved in any decision and their relationship. The information and tools available in the Web are key factors nowadays to make better decisions as they are used by main agriculture actors in for example planning activities. This article presented a novel approach which relies on different techniques for assisting the user's needs (traditional known as requirement gathering task) based on exploratory studies and empathy task as well as generating solutions built on top of existing Web apps. By combining Web Augmentation techniques with Model-Driven Web Engineering approaches, the solution only requires to be idealized and designed and source code is generated without requiring a software developer intervention. To illustrate the approach, introduced a case study which involved performing a design thinking experience in a farm located in La Plata, Argentina was introduced. The outcome of the study was a tool that helps farmers, and procurement agents to gather information of raw material, collaboratively evaluate the alternatives and finally assist the decision making task.

Innovation in information systems for agriculture is needed. Information systems have the potential to help stakeholders deal with the stressing conditions imposed by climate and market change, and to embrace sustainable practices. However, innovation is a domain where information systems are still maturing (such as agriculture) is challenging. This work showed that Design Thinking, Web Augmentation, and Collaboration Engineering are effective tools to foster innovation in agriculture.

Further work involves conducting an experiment to assess the effectiveness and productivity of this approach. There are also plans to extend the scope of the application to other business domain such as financial and retail.

## Acknowledgement

## References

1. R. H. Sprague Jr. and E. D. Carlson, *Building Effective Decision Support Systems*. Prentice Hall Professional Technical Reference, 1982.
2. G. M. Marakas, *Decision support systems in the 21st century*, Second Edi., vol. 134. Prentice Hall Upper Saddle River, NJ, 2003.
3. S. Fountas *et al.*, "Farm management information systems: Current situation and future perspectives," *Comput. Electron. Agric.*, vol. 115, pp. 40–50, Jul. 2015.
4. M. J. Blackie, "Management information systems for the individual farm firm," *Agric. Syst.*, vol. 1, no. 1, pp. 23–36, 1976.
5. M. J. Carrer, H. M. de Souza Filho, and M. O. Batalha, "Factors Influencing the Adoption of Farm Management Information Systems (FMIS) by Brazilian Citrus Farmers," *Comput. Electron. Agric.*, vol. 138, no. C, pp. 11–19, Jun. 2017.
6. J. Alvarez and P. Nuthall, "Adoption of computer based information systems: The case of dairy farmers in Canterbury, NZ, and Florida, Uruguay," *Comput. Electron. Agric.*, 2006.
7. H. A. Simon, *The New Science of Management Decision*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1977.
8. R. Wang and D. Strong, "Beyond accuracy: What data quality means to data con-sumers," *J. Manage.*, 1996.
9. J. H. E. Hernandez *et al.*, "RUC-APS: Enhancing and implementing Knowledge based ICT solutions within high Risk and Uncertain Conditions for Agriculture Production Systems," in *11th International Conference on Industrial Engineering and Industrial Management*, 2017.
10. T. Brown, "Change by design," 2009.
11. N. O. Bouvin, "Unifying strategies for Web augmentation," in *Proceedings of the tenth ACM Conference on Hypertext and hypermedia: returning to our diverse roots: returning to our diverse roots*, 1999, pp. 91–100.
12. O. Díaz and C. Arellano, "The Augmented Web: Rationales, Opportunities, and Challenges on Browser-Side Transcoding," *TWEB*, vol. 9, no. 2, p. 8, 2015.
13. D. Firmenich, S. Firmenich, J. M. Rivero, L. Antonelli, and G. Rossi, "Crowd-Mock: an approach for defining and evolving web augmentation requirements," *Requir. Eng.*, vol. 23, no. 1, pp. 33–61, 2018.
14. O. Díaz, C. Arellano, I. Aldalur, H. Medina, and S. Firmenich, "End-User Browser-Side Modification of Web Pages," in *Web Information Systems Engineering -- WISE 2014*, 2014, pp. 293–307.

15. I. Aldalur, M. Winckler, O. Díaz, and P. Palanque, "Web Augmentation as a Promising Technology for End User Development," in *New Perspectives in End-User Development*, F. Paternò and V. Wulf, Eds. Cham: Springer International Publishing, 2017, pp. 433–459.

16. J. E. Hernandez *et al.*, "Challenges and Solutions for Enhancing Agriculture Value Chain Decision-Making. A Short Review BT - Collaboration in a Data-Rich World," 2017, pp. 761–774.

17. P. Zaraté, G. Camilleri, and D. M. Kilgour, "Multi-criteria Group Decision Making with Private and Shared Criteria: An Experiment," in *International Conference on Group Decision and Negotiation*, 2016, pp. 31–42.

18. E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design patterns: elements of reusable object-oriented software*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1995.

19. T. Schummer and S. Lukosch, *Patterns for computer-mediated interaction*. John Wiley & Sons, 2013.

20. R. O. Briggs, G.-J. De Vreede, J. F. Nunamaker, and D. Tobey, "ThinkLets: achieving predictable, repeatable patterns of group interaction with group support systems (GSS)," in *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*, 2001, p. 9--pp.

21. P. B. Lowry and J. F. Nunamaker, "Using the thinkLet framework to improve distributed collaborative writing," in *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on*, 2002, pp. 272–281.

22. R. O. Briggs, G.-J. De Vreede, and J. F. Nunamaker Jr, "Collaboration engineering with ThinkLets to pursue sustained success with group support systems," *J. Manag. Inf. Syst.*, vol. 19, no. 4, pp. 31–64, 2003.

23. M. Urbieta, S. Firmenich, P. Maglione, G. Rossi, and M. A. Olivero, "A Model-driven Approach for Empowering Advance Web Augmentation - From Client-side to Server-side Support," in *APMDWE*, 2017.

24. S. Firmenich, G. A. Bosetti, G. Rossi, M. Winckler, and T. Barbieri, "Abstracting and Structuring Web Contents for Supporting Personal Web Experiences," in *Web Engineering - 16th International Conference, {ICWE} 2016, Lugano, Switzerland, June 6-9, 2016. Proceedings*, 2016, pp. 77–95.

25. A. Popovici, T. Gross, and G. Alonso, "Dynamic Weaving for Aspect-oriented Programming," in *Proceedings of the 1st International Conference on Aspect-oriented Software Development*, 2002, pp. 141–147.

26. "Censo hortifloricola de la provincia de buenos aires," 2005.

27. S. Firmenich *et al.*, "Engineering concern-sensitive navigation structures, concepts, tools and examples," *J. Web Eng.*, vol. 9, no. 2, pp. 157–185, 2010.