Farmbot, a Small Scale Autonomous Farming Machine: Software Challenges

Johann Bourcier, Univ. Rennes 1, France
Olivier Barais, Univ. Rennes 1, France
Nicolas Harrand, Univ. Rennes 1, France
Alexandre Rio , Univ. Rennes 1, France
Benoit Combemale, Univ. Toulouse, France

Key words: Farmbot, Resource Aware, Domain specific Optimization

1 Farmbot

A Farmbot[1] is a small scale fully automated farming system. The FarmBot is a complete open source project providing an easy access to small scale local farming. The current size of the farming landscape is 1.5 meter wide by 3 meters long. Using this open-source project, family or professional are able to automatically grow their own vegetable by automating all the repetitive tasks of precise seeds planting and watering, removing weeds, etc. Figure 1 shows a working prototype of the farmbot project.

Technically, a Farmbot system can be seen as a farming Computer Numerical Control (CNC)[2] machine. The Farmbot system embeds two main computing units: an embedded computer (a raspberry Pi), and a micro-controller (an Arduino mega). From a software perspective, the Arduino hosts software to directly control the motors, and all actuators and sensors of the system. It provides an API reusing standard protocols over a serial link. The raspberry Pi is in charge of automating, coordinating and scheduling all the tasks of the farming machine which are then delegated to the Arduino. The raspberry Pi is also in charge of image processing to detect weeds, and provides an API to discuss with the cloud hosted software part. Finally, a web application is hosted on the cloud and provides connection to external knowledge such as the crop growing needs, the local weather forecast, the decision making support, and a web user interface.

Figure 1: A Farmbot prototype (*photo credit FarmBot team*)

2 On Functional Extensions of the Farmbot

At the University of Rennes 1 (France), we recently bought a Farmbot system with two primary objectives: first to experiment with students the design and evolution of connected objects, and second to experiment various software engineering research techniques we are currently investigating. To this purpose, we envision the design of the three following main functional extensions of the Farmbot system.

Off grid resources management: In an off grid scenario, resources such as electricity and water are not infinite. As their production (rain collection, solar panel) depends on external factors, efficient storage (reservoir and battery) and consumption decisions must be taken accordingly. It requires both anticipations to evaluate

[1] Cf. `https://farmbot.io/`

[2] See `https://en.wikipedia.org/wiki/Numerical_control` for more information

the quantity of resources that will be produced and consumed in the next hours, days or weeks, and quick reactions to decide on corrective actions when the measured reality deviates from the plan.

Permaculture advisor: The Farmbot have both an access to data collected on the field, and knowledge gather by the community. Thus, agronomic domain specific advices could be formulated to the user, matching theoretical models to empirical data. This could assist the user in its choice of crop combination, layout and cycles, thereby helping to design a permaculture[3].

Automatic malfunction or disease detection: In such a system, various failures and hazard may happen. We propose to leverage on sensing capabilities (camera and other) of the Farmbot to detect events or situations that deviate from a norm in order to alert users, and thus prevent further damages to crops. This situation could include crop diseases, Farmbot malfunctions or external hazard.

3 Software Challenges for the Farmbot Extensions

The design of the functional extensions aforementioned requires specific efforts from the software engineering research community to address the underlying challenges. In this section, we describe three of these main research challenges.

Designing resource aware software: The design of a resource aware software requires an approach where the software will adapt its functional and non-functional behavior dynamically according to the available resources. The general problem for software engineering is therefore the automatic optimization of resource usage for a software which requires **seamless combination of proactive and reactive adaptation** of the software behavior.

In the particular case of the Farmbot system, we propose to investigate a smart resources manager that will schedule tasks consuming these resources (network access, energy and water) based on both the actual stocks and prevision of production. Leveraging the Farmbot access to weather forecast and its embedded sensors, the embedded software will become aware of the available resources and the prediction of resource productions. Taking into account a specific model that will be built by monitoring its normal activity, the system will be able to build a model of resource production and consumption to optimize the resource usage.

Designing domain specific optimization Optimizing food production and vegetable growing is not part of the expertise of a computer scientist. Nevertheless, the software in charge of automating and optimizing the Farmbot operations has to leverage this knowledge to both assist the user decision making when choosing the seeds layout, and when automating the plant watering.

The general problem relates to the use of heterogeneous models and their combination in order to optimize the operations of a software system which controls physical mechanisms (such as Cyber Physical Systems, or Internet of Things devices). A particular instantiation of this problem on the Farmbot system would be to leverage knowledge about multi crop system, weather forecast, biomass growing and so on to optimize the layout of seeds and their mutual benefit to optimize the food production.

Complex event detection through multi sensors stream analysis: The automatic detection of malfunctions or disease is a very complex and challenging task since it requires to crosscheck and match data coming from different sensors such as humidity, images, and other chemical soil characteristics. Furthermore, the instantaneous value of these sensors may not be sufficient to detect malfunctioning. The study of their evolution over time will give significantly better results on such detection.

The general challenge for software engineering is therefore related to the ability to perform complex relations and correlations between values produced by various sensors at different locations and their evolution over time. In the specific case of the Farmbot system, it means correlating values from the evolution of the soil characteristics at various locations, and image analysis to detect the size, holes in leafs and so on.

4 Conclusion

The Farmbot system is a very interesting connected object which has a societal impact through a paradigm shift for local food production. It also has many interests from a research perspective in the software engineering community since it represents a fantastic experimental case study for various open research problems in modern heterogeneous and distributed software which interact directly with our physical environment.

In particular we draw in this paper several possible extensions of the Farmbot that raise software engineering challenges, such as the design and integration of different domain-specific languages, the composition of various kinds of models, and the computation of complex relations and correlations from such a set of models.

[3]Permaculture is a system of agricultural and social design principles centered around simulating or directly utilizing the patterns and features observed in natural ecosystems (Wikipedia)