A Decentralized Decision Support System for Mobile Devices

Gert Scholten, Nicholas Palmer, Roelof Kemp, Thilo Kielmann, and Henri Bal

Vrije Universiteit
De Boelelaan 1081A, 1081 HV, Amsterdam, The Netherlands
gscholt@gmail.com,
{palmer,rkemp,kielmann,bal}@cs.vu.nl

Abstract. With this demonstration we present *Decisionlib*, a Distributed Decision Support System that runs on mobile devices. Decisionlib is a flexible, durable, and robust voting system which places emphasis on communication and error handling in order to provide reliable and easy to program group-based decisions on distributed mobile devices. Written on top of the *Ibis* [4] distributed communication system, Decisionlib represents the state of the art for group decision making on mobile devices.

Group based decisions generally involve reaching some form of agreement between a number of different participants. This kind of decision is vital in the disaster management field where leadership roles are often not well defined[2] and quick decisions need to be made by people in the field based on varying conditions they inevitably encounter in order to provide the agility[1] required for an effective response.

To illustrate the problems that arise in mobile, distributed group based decisions we present the following scenario from the disaster management literature[3]; A major disaster such as an earthquake or a hurricane landfall and there is destruction on a massive scale leaving the remaining survivors stranded. Rescue workers and non-injured effected individuals search the area, collect groups of survivors and provide immediate first aid. For this process to work smoothly, people need to be grouped so that efficient aid can be provided, and people can be easily transported out of the disaster area. Rescue workers and effected people find each other by use of GPS locations, using ad-hoc networking via Wi-Fi and Bluetooth and mark their preference for a gathering point. The gathering points are evaluated and voted upon among those in the area while some participants lose connectivity. Nonetheless, a reachable location is selected to bring everyone together for faster evacuation. Shortly after the decision is made users who lost connectivity during the decision process reconnect to the network and receive notification of the chosen gathering point.

We have identified the following requirements for a distributed decision support system for disaster situations:

M. Griss and G. Yang (Eds.): MOBICASE 2010, LNICST 76, pp. 302–305, 2012.

Flexible: The system should be highly configurable to allow users to specify what kind of decisions to make, and how to make them.

Durable: The system should provide durable results. In the scenario above, some participants lose network connectivity. This should not have a permanent impact on the availability of the result of a decision. Optionally, entities not actively participating in the decision should be made aware of the result. This allows for partial participation in decisions, and public versus private decisions.

Ubiquitous: The system should allow for decisions to be easily made while out in the field and preferably already available on devices which are normally carried by people likely to be effected by disasters.

Robust: The system should manage failures in a configurable manner and should not rely on central components, because these are prone to failure, especially in disaster scenarios.

To summarize, we need a flexible system on mobile phone which handles failures well, provides durable results and runs without relying on centralized components.

In this demo we present the Decisionlib Distributed Decision Support System which satisfies all of these requirements. It is configurable to support any kind of decision process and to provide the information required for group based decision making. It deals with connectivity issues in ways that do not hinder the reaching of a decision. A decision is considered final and possibly universally available after it has been made, meeting all of the above requirements.

The demonstration of this system will be run using a number of Android powered Smartphones. No additional materials are required from the Venue. One of the phones will be used in Ad-Hoc mode to provide a network connection

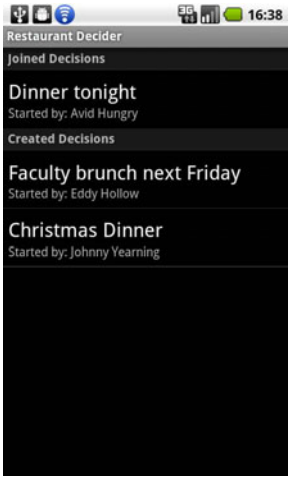

Fig. 1. List of Decisions

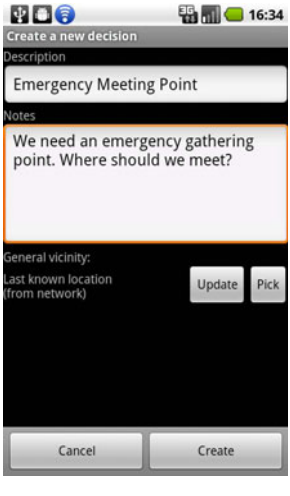

Fig. 2. Create a New Decision

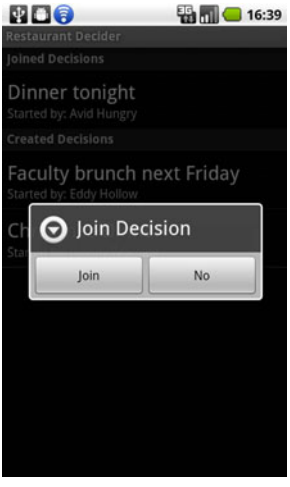

Fig. 3. Join a Decision

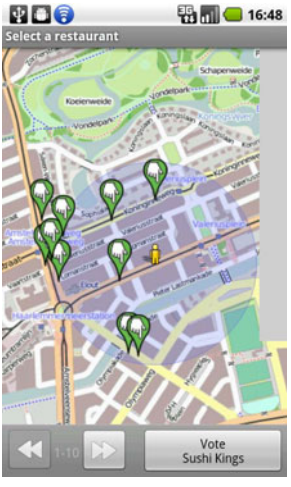

Fig. 4. Free Vote On A Map

between the phones demonstrating that our system can be used in a fully decentralized, ad-hoc environment. During the demonstration participants will be invited to use one of our phones to participate in a decision making process. For this demonstration participants will be invited to select a restaurant near the conference venue at which to meet other participants. Note that this process is completely analogous to the situation presented in our disaster management example above but is of more direct interest to conference participants. Users will be invited to create a decision, as shown in Figure 2 and select the participants,

Fig. 5. Vote List

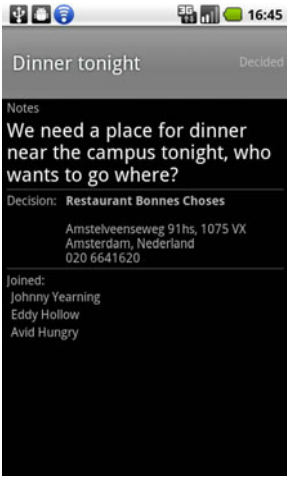

Fig. 6. View Completed Decision

to view a past decision as shown in Figure 1, or join a currently running decision as shown in Figure 3. In the first round of voting users will be allowed to select a restaurant from a map with local venues as shown in Figure 4. Voting then proceeds in rounds using a list of selected restaurants as seen in Figure 5. Once a decision is reached it is available for viewing as shown in Figure 6. Note that the fact that our application can be used for a number of every day scenarios means that it is much more likely to be deployed and users will already know how to use it when a disaster strikes.

Conclusion

This demonstration features the *Decisionlib* Distributed Decision Support System. Our analysis of disaster management literature led us to the following requirements for a distributed decision support library: flexible, durable, ubiquitous and robust. The Decisionlib decision support system we have built meets these requirements by supporting reliable communication, membership management, recovery in case of connection failures and persistence in decisions and their results as long as a participating node is alive. It supports many types of decisions, and any round-based voting algorithm. It comes with several predefined algorithms and application designers are free to add their own or tailor an existing algorithm to meet the needs of the application. It runs on smartphones which people are likely to carry with them everywhere.

Our Restaurant Decision test application was used to evaluate the utility of the Decisionlib system. The application is a reliable application for deciding on temporary gathering points, and uses only a few lines of code to use the Decisionlib system. It illustrates the ease of writing an application on top of the Decisionlib, and furthermore demonstrates that it works reliably in hostile situations such as those encountered as part of disaster management.

References

1. Harrald, J.: Agility and Discipline: Critical Success Factors for Disaster Response. The ANNALS of the American Academy of Political and Social Science 604(1), 256 (2006)
2. Plotnick, L., Ocker, R., Hiltz, S., Rosson, M.B.: Leadership roles and communication issues in partially distributed emergency response software development teams: A pilot study. In: HICSS, vol. 29 (2008)
3. U.S.C.S.B.C. to Investigate the Preparation for, R. to Hurricane Katrina, T. Davis, and U. S. G. A. Office: A Failure of Initiative: Final Report of the Select Bipartisan Committee to Investigate the Preparation for and Response to Hurricane Katrina. Govt. Printing Office (2006)
4. van Nieuwpoort, R.V., Maassen, J., Wrzesinska, G., Hofman, R., Jacobs, C., Kielmann, T., Bal, H.E.: Ibis: a flexible and efficient Java based grid programming environment. Concurrency and Computation: Practice and Experience 17(7-8), 1079–1107 (2005)