

# Parliament: A Module for Parliamentary Procedure Software

BAYLE SHANKS AND DANA DAHLSTROM

## 1 Introduction

Parliament is an open source software module that can be used to build programs that follow or moderate the conduct of a deliberative assembly using parliamentary procedure. Parliament encapsulates logic and bookkeeping functions necessary for the function of parliamentary procedure and can be embedded in applications for face-to-face meetings or for synchronous or asynchronous computer mediated communication.

Parliament's central functions track meeting state aspects of the meeting, such as pending motions, the relationships among them, and business already transacted. The outer application (see Figure 1) is responsible for informing the Parliament module about events as they occur in the meeting, such as 'member X made motion Y' or 'motion Y failed.' Parliament answers queries such as 'Which motions are presently valid?' or 'Which motions have been adopted in this meeting?' It is also capable of answering questions about hypothetical situations, such as 'Which motion will be pending if the immediately pending motion is adopted?' The Parliament module does not incorporate the details of parliamentary procedure, such as the motions and customs described in *Robert's Rules of Order Revised* (Robert 1915). Instead, Parliament requires an external rule specification, allowing the user or developer to modify the rules independently and even to develop whole new rule systems.

*Online Deliberation: Design, Research, and Practice.*  
Todd Davies and Seeta Peña Gangadharan (eds.).  
Copyright © 2009, CSLI Publications.

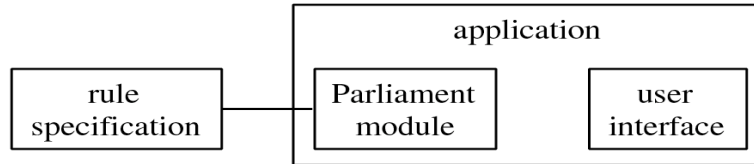


Figure 1. The Parliament module is embedded in an application, and uses an external rule specification<sup>1</sup>

## 2 A Reusable Model

Many software applications could share a common software implementation of parliamentary procedure; for example, applications that:

- (for online, synchronous meetings)
  - assist meeting participants during the meeting
  - assist meeting officers during the meeting
  - train people before the meeting
  - provide a networked application which participants use to request the floor and to make and vote on motions
- (for online, asynchronous meetings—Web or other)
  - assist a human chair
  - automatically chair a meeting
  - moderate a large discussion board or wiki according to formal meeting rules
  - automatically update a set of organizational bylaws according to the instructions of an online deliberative assembly

We offer the Parliament module for use in other programs (applications), since it would be inefficient to reimplement the core logic of parliamentary rules and sets of motions for each software project.

## 3 Modular Rule Specifications

*Robert's Rules of Order Revised* is the most common parliamentary authority in the United States, but there are others. Older, public domain versions of *Robert's Rules of Order* and Sturgis's *Standard Code of Parliamentary Procedure* are examples. Each branch of the U.S. Congress uses its own rules, which are broadly similar to others just mentioned.

Rather than choosing a specific set of parliamentary rules and 'hard-coding' them, we designed a single module to accommodate many different parliamentary rule sets. The user or developer may describe new rule sets in a specification language understood by the Parliament module, and the

specification is then loaded at runtime (see Figure 1 above). There are many advantages to this approach:

- Different deliberative assemblies use different meeting rules.
- Unconventional meeting settings such as the Web or Internet relay chat (IRC) demand new innovations in parliamentary rules.
- Allowing the rule set to be modified gives each assembly complete flexibility to adapt the software to its needs. Assemblies should not be forced to follow a particular set of meeting rules just because their software cannot support the rules that they prefer.
- Research on group decision making support systems (GDSS) is hindered by the difficulty of isolating the effect of individual components of the group decision making process. A configurable parliamentary rule set will serve as the ideal platform for testing fine-grained modifications to a group's process.

#### **4 The Rule Specification Language**

The rule specification language has a quasi-English syntax. A rule set is typically written in a separate file and then loaded into the Parliament module upon initialization.

The rule set is specified in terms of ‘actions’ that participants can take at certain times in the meeting.

The rule set specification can be arbitrarily expressive: if there is no other way to express some desired behavior, Python code can be embedded into the rule set.

#### **5 The Robert's Rules Parliamentary Rule Set Specification**

We have written a partial rule set specification for the 1915 (now public domain) fourth edition of *Robert's Rules of Order Revised*. The rule set includes over twenty-five of the most common motions and their important attributes—such as when they are debatable and what vote is required for them to carry—as well as most of the precedence relations between the motions and some of their semantics.

This specification was initially based on Henry Prakken's formalization of Robert's Rules (Prakken 1998), which he kindly provided to us in machine-readable form. We made many changes to Prakken's formalization, including the addition of the complicated logic of precedence.

## 6 Conclusion

The Parliament module provides the central infrastructure for parliamentary-procedure software. A reusable module, it implements and interprets parliamentary rules, tracks meeting state, and infers important information such as which motions are in order at a given time. The module is flexible, accommodating any properly codified set of parliamentary rules.

```

NAME: Lay on the table
MOTION TO FORM OF NAME: "Motion to lay on the table"

TYPE: Subsidiary motion

SUMMARY: "The objective of this motion is to temporarily lay a question aside"

motion precedence: 1
debatable: NO
amendable: NO
subsidiaries allowed: NO
reconsiderable: ONLY WHEN (WAS_ACCEPTED)

TARGET: ancestor motion
ON PASS: table target

category: "scheduling"
purpose: "delay"

# comments can be embedded like this

RRO section ref: "19"

RROR section ref: "28"

{
def example_method(self):
print 'This is embedded Python code'
}

```

Figure 2. The motion to ‘Lay on the table’ defined in the rule specification language (note that in the actual code there would be no word-wrapping)

Figure 2 shows the definition of an example motion in the rule specification language, taken from the Robert's Rules rule set.

A concise specification language allows others to create and modify rule sets efficiently. A usable partial specification of Robert's rules has been created, and a prototype Robert's rules meeting assistant has been built and used in real face-to-face meetings (Dahlstrom and Shanks 2009).

The Parliament module shows potential for use in many contexts, including both face-to-face and online meetings. It is hoped the module will lead to a variety of useful parliamentary software.

## References

- Dahlstrom, D. and B. Shanks. 2009. Software Support for Face-to-Face Parliamentary Procedure. *Online Deliberation: Design, Research, and Practice*, eds. T. Davies and S. P. Gangadharan, 213-220. Stanford, CA: CSLI Publications.
- Prakken, H. 1998. Formalizing Robert's Rules of Order: An Experiment in Automating Mediation of Group Decision Making. *Tech. Rep. REP-FIT-1998-12*, GMD.
- Robert, H. M. 1915. *Robert's Rules of Order Revised*. Public Domain.
- Sturgis, A. 2001. *The Standard Code of Parliamentary Procedure*, 4<sup>th</sup> ed. New York: McGraw-Hill.

