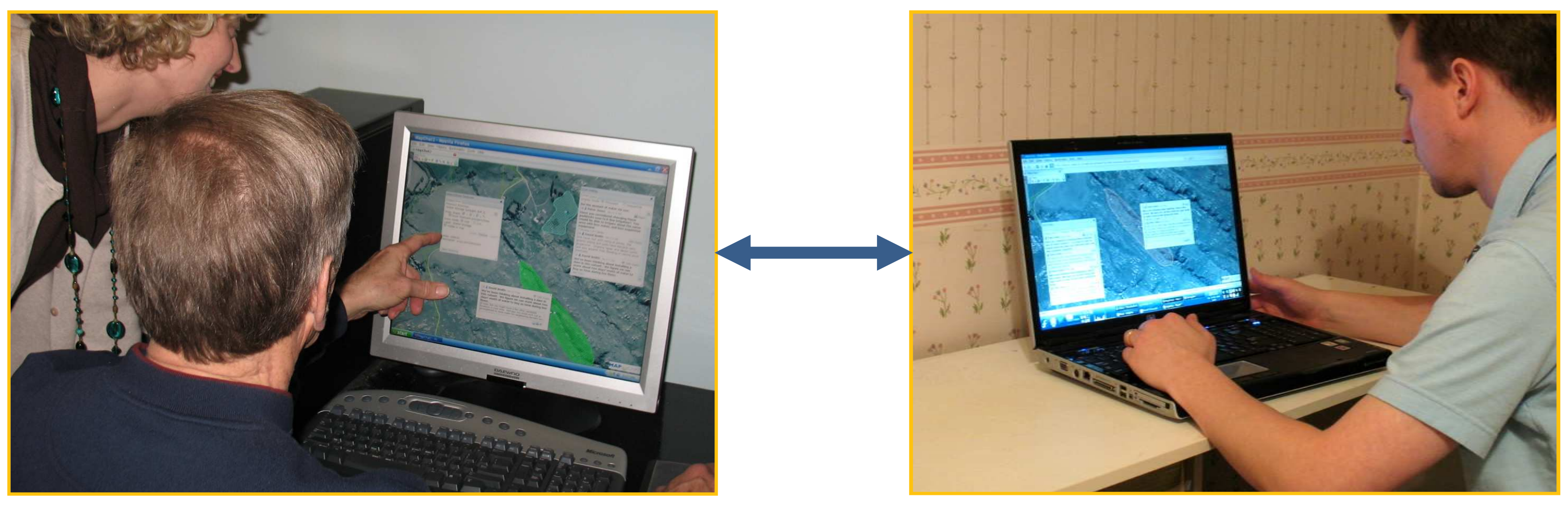
MAP CHAT

Multi-user Web Mapping with Open Source Components: MapChat Version 2.0

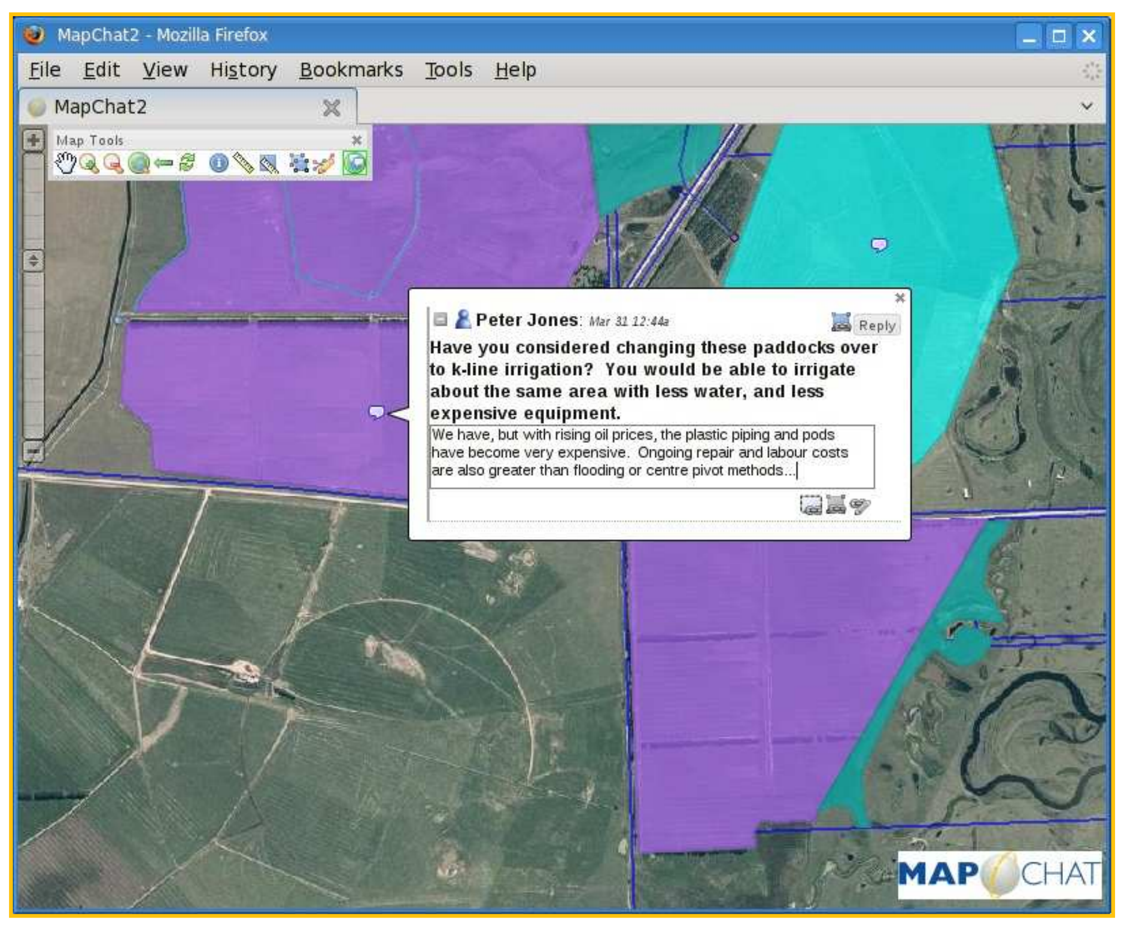
Overview:

MapChat is a Web-based mapping application designed to allow multiple people to use an online map as a collaborative tool for discussing spatially-sensitive issues. The approach used by MapChat supplements the 'geotagging' concepts popularized by Google Earth/Google Maps and similar tools with 'geochatting'. The objective of geochatting is to facilitate content and knowledge mapping as well as social networking to exchange views on issues of local importance.



Geochatting:

The geochatting concept is operationalized in MapChat with a Web-map interface that forms the medium of communication between users. Customized, synchronous instant messaging facilitates collaborative discussion between dispersed participants, where each participant can geotag map features and communicate, or chat, via private and public messaging with other individuals involved in a discussion.



MapChat2 - Mozilla Firefox

Peter Jones: Mar 31 12:44a Reply

Have you considered changing these paddocks over to k-line irrigation? You would be able to irrigate about the same area with less water, and less expensive equipment.

We have, but with rising oil prices, the plastic piping and pods have become very expensive. Ongoing repair and labour costs are also greater than flooding or centre pivot methods...

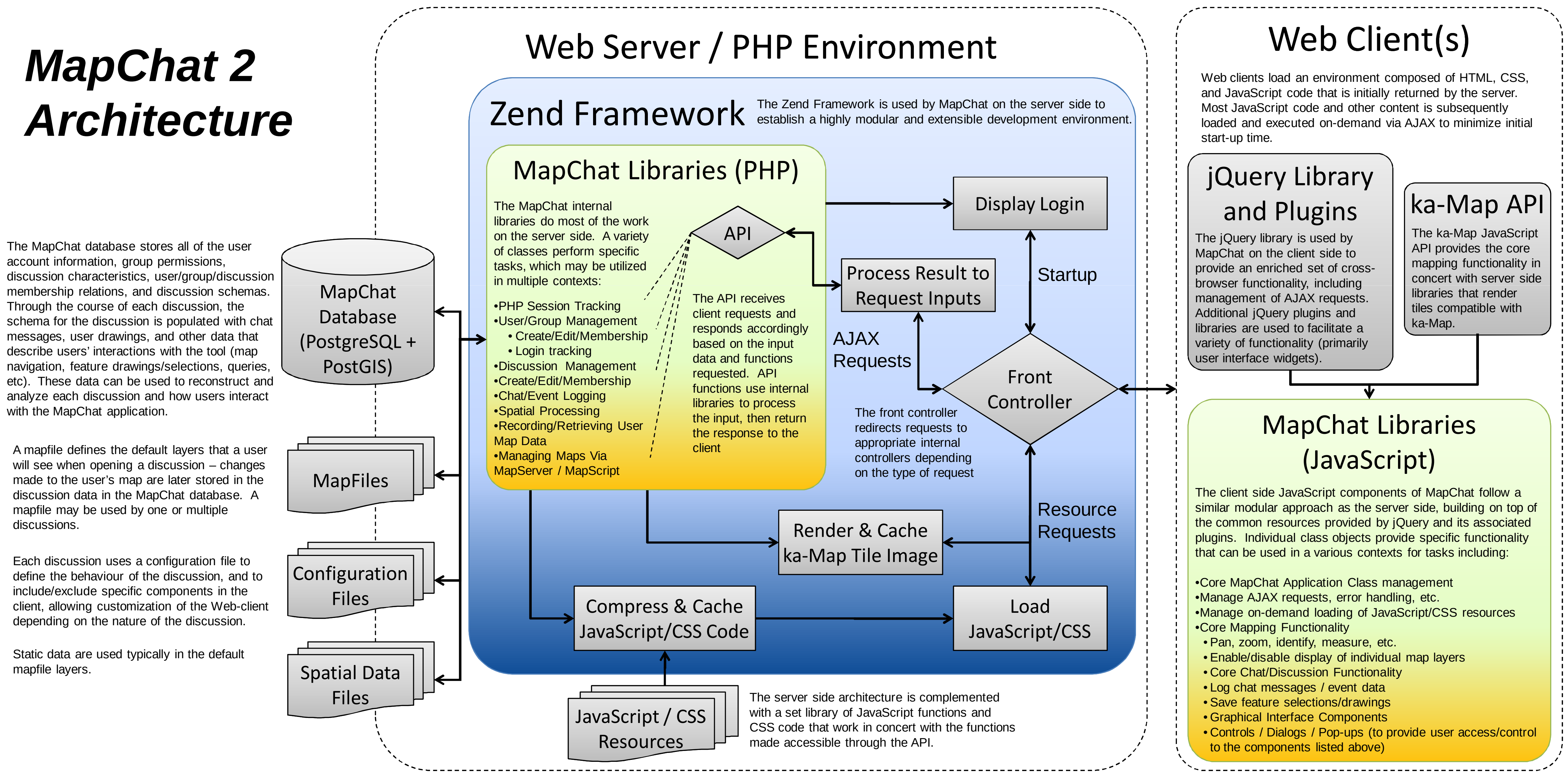
Example of GeoChatting as implemented in MapChat 2

Technical Re-design of MapChat:

MapChat 2 is significantly improved from its original version. While the overall concept remains the same, much more attention is given to the organization of code and data structure.

Version 2 is redesigned from the ground-up to take better advantage of AJAX techniques, on-demand loading/compression of code, and many other improvements. The result is a Web interface that is much more dynamic (but also usable even over dial-up connections), a much more usable/extendible code base, and more detailed/informative data collection. This modular implementation allows components to be included or excluded in the interface for different discussions as they are needed or developed.

MapChat 2 Architecture



Web Server / PHP Environment

Zend Framework — The Zend Framework is used by MapChat on the server side to establish a highly modular and extensible development environment.

MapChat Libraries (PHP)

The MapChat internal libraries do most of the work on the server side. A variety of classes perform specific tasks, which may be utilized in multiple contexts:

- •PHP Session Tracking
- •User/Group Management
  - Create/Edit/Membership
  - Login tracking
- •Discussion Management
- •Create/Edit/Membership
- •Chat/Event Logging
- •Spatial Processing
- •Recording/Retrieving User Map Data
- •Managing Maps Via MapServer / MapScript

API

The API receives client requests and responds accordingly based on the input data and functions requested. API functions use internal libraries to process the input, then return the response to the client

Display Login
Process Result to Request Inputs
Startup
AJAX Requests
Front Controller
Resource Requests
Render & Cache ka-Map Tile Image
Compress & Cache JavaScript/CSS Code
Load JavaScript/CSS
MapChat Database (PostgreSQL + PostGIS)
MapFiles
Configuration Files
Spatial Data Files
JavaScript / CSS Resources

The front controller redirects requests to appropriate internal controllers depending on the type of request

The MapChat database stores all of the user account information, group permissions, discussion characteristics, user/group/discussion membership relations, and discussion schemas. Through the course of each discussion, the schema for the discussion is populated with chat messages, user drawings, and other data that describe users' interactions with the tool (map navigation, feature drawings/selections, queries, etc). These data can be used to reconstruct and analyze each discussion and how users interact with the MapChat application.

A mapfile defines the default layers that a user will see when opening a discussion – changes made to the user's map are later stored in the discussion data in the MapChat database. A mapfile may be used by one or multiple discussions.

Each discussion uses a configuration file to define the behaviour of the discussion, and to include/exclude specific components in the client, allowing customization of the Web-client depending on the nature of the discussion.

Static data are used typically in the default mapfile layers.

The server side architecture is complemented with a set library of JavaScript functions and CSS code that work in concert with the functions made accessible through the API.

Web Client(s)

Web clients load an environment composed of HTML, CSS, and JavaScript code that is initially returned by the server. Most JavaScript code and other content is subsequently loaded and executed on-demand via AJAX to minimize initial start-up time.

jQuery Library and Plugins — The jQuery library is used by MapChat on the client side to provide an enriched set of cross-browser functionality, including management of AJAX requests. Additional jQuery plugins and libraries are used to facilitate a variety of functionality (primarily user interface widgets).

ka-Map API — The ka-Map JavaScript API provides the core mapping functionality in concert with server side libraries that render tiles compatible with ka-Map.

MapChat Libraries (JavaScript)

The client side JavaScript components of MapChat follow a similar modular approach as the server side, building on top of the common resources provided by jQuery and its associated plugins. Individual class objects provide specific functionality that can be used in a various contexts for tasks including:

- •Core MapChat Application Class management
- •Manage AJAX requests, error handling, etc.
- •Manage on-demand loading of JavaScript/CSS resources
- •Core Mapping Functionality
  - Pan, zoom, identify, measure, etc.
  - Enable/disable display of individual map layers
  - Core Chat/Discussion Functionality
  - Log chat messages / event data
  - Save feature selections/drawings
  - Graphical Interface Components
  - Controls / Dialogs / Pop-ups (to provide user access/control to the components listed above)

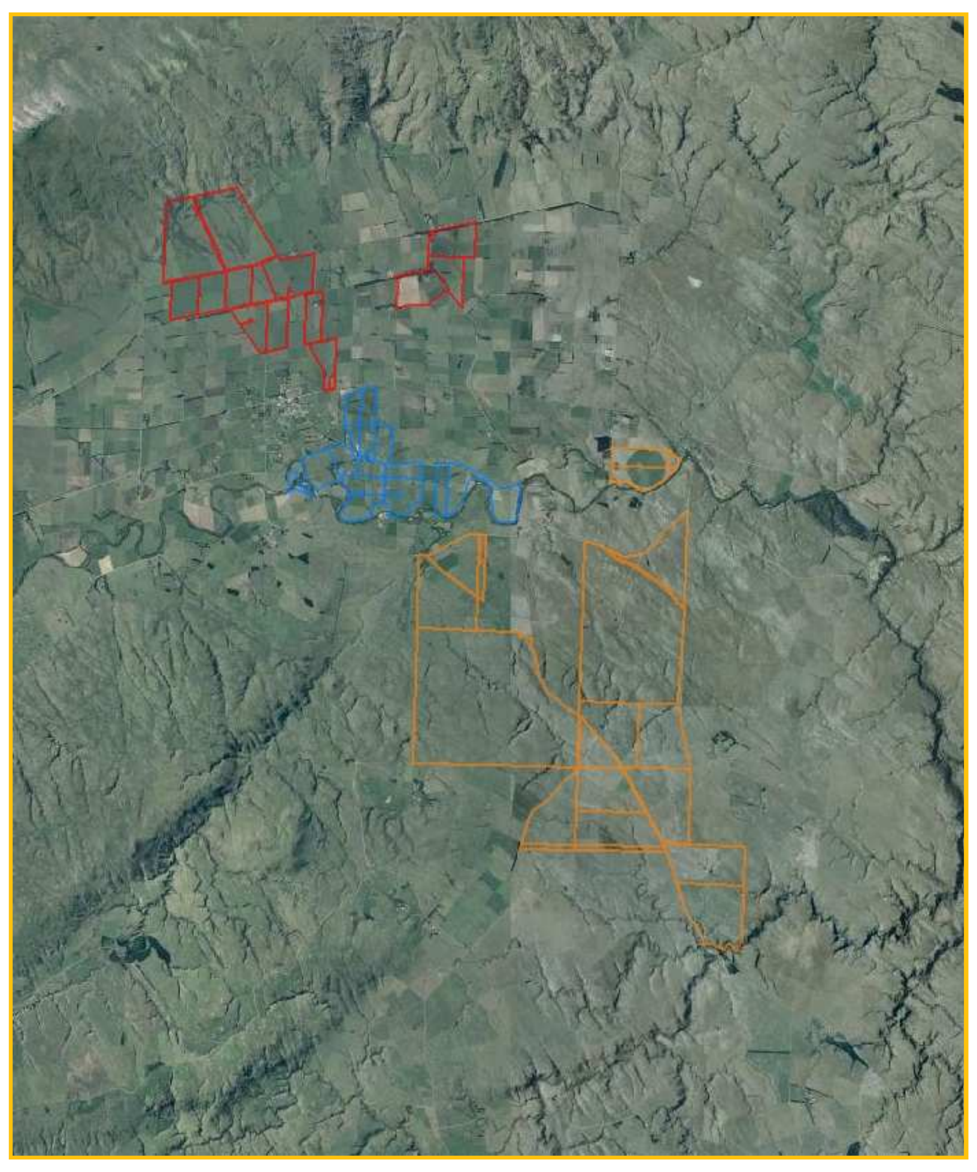
The design of MapChat 2 breaks away from the previous version by using a set of objects and classes designed specifically for version 2. The server side is on built top of the Zend Framework (http://framework.zend.com) for PHP, while the client-side utilizes the jQuery (http://jquery.com) JavaScript library and associated plugins. The mapping components are based on the ka-Map (http://ka-map.maptools.org) server side and client side APIs. The remaining underlying components include PostgreSQL (http://www.postgresql.org) and PostGIS (http://www.postgis.org) for the spatial database, and MapServer/MapScript (http://www.mapserver.org) for rendering map images and managing mapfiles. All of MapChat's components are derived from Free/Libre and Open Source Software (FLOSS).

Overlooking the Paerau Weir (foreground) and the Maniototo Plains (background). Many farmers in this area of the Upper Taieri River catchment are members of a successful private irrigation company – an example of what could be implemented for the rest of the catchment.

Case Study – Upper Taieri River Catchment, New Zealand:

MapChat 2 was tested in a rural community in the Upper Taieri River catchment in the South Island of New Zealand. Water in this catchment is a scarce resource that is essential for traditional agricultural practices dominated by dry land mixed sheep and beef farming. As farmers in this area face a changing national and global economy, changing environmental conditions, and new regulations on the use of water, they are seeking ways to cooperate at a community level to help ensure the security of their local water supply.

Phased Case Study Approach:

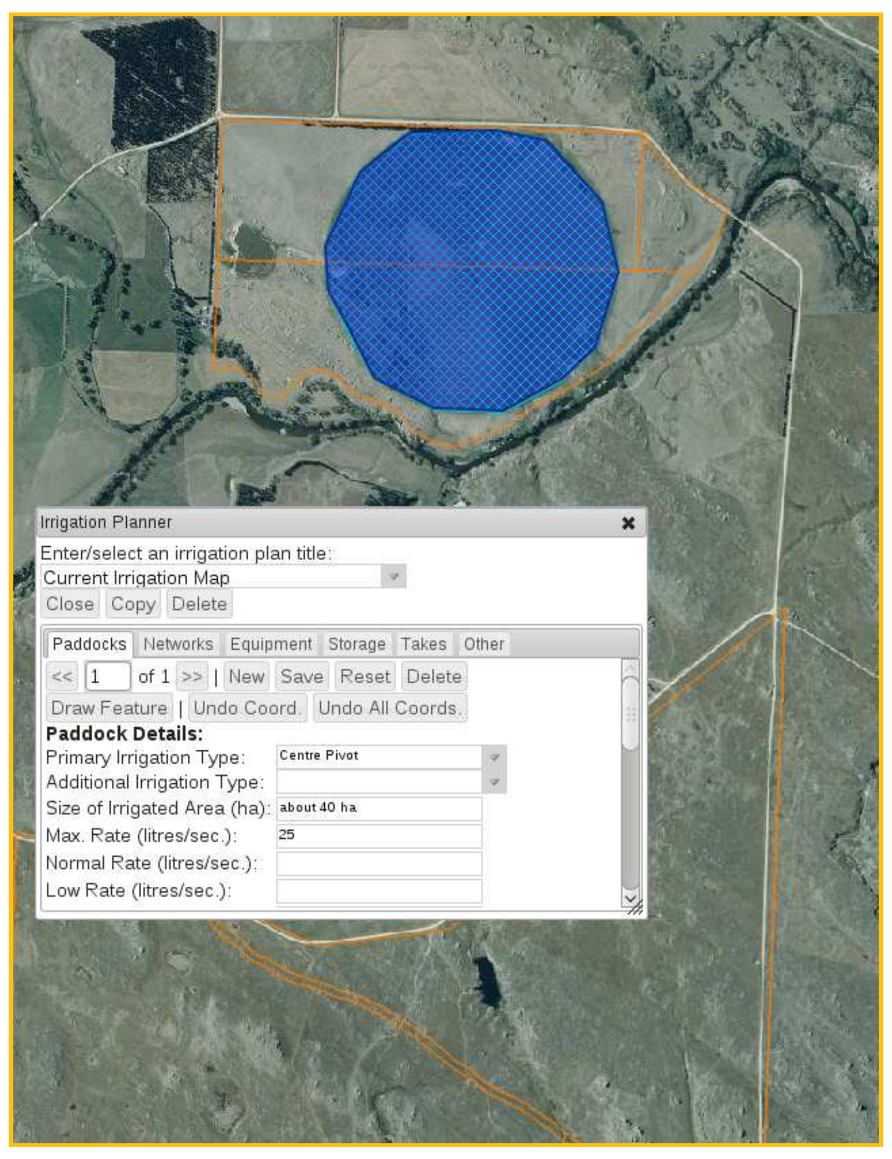


Sample of boundaries for three farm management units identified for participants in the Upper Taieri catchment.

Phase 1: As volunteers for the case study were identified, they were initially asked to identify the property boundaries for each of their farms.

Phase 2: Each participant was given access to a tool in MapChat for individually mapping their use of water for irrigation. In some cases, the participants were given the option to perform this phase unassisted after being given an instructional tutorial. The rest were directly assisted by a researcher to complete the task.

Data collected during the first two phases of the case study provided the contextual map data needed for group discussions in the following phases.



Irrigation Planner

Enter/select an irrigation plan title: Current Irrigation Map

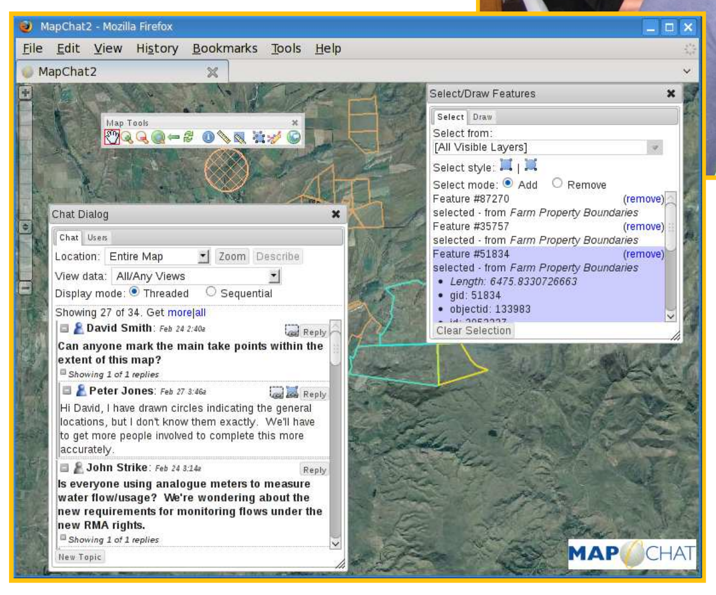
Close
Copy
Delete
Paddocks
Networks
Equipment
Storage
Takes
Other
<< 1 of 1 >> | New Save Reset Delete
Draw Feature | Undo Coord. Undo All Coords.

Paddock Details:

Primary Irrigation Type: Centre Pivot
Additional Irrigation Type:
Size of Irrigated Area (ha): about 40 ha
Max. Rate (litres/sec.): 25
Normal Rate (litres/sec.):
Low Rate (litres/sec.):

Example of the irrigation mapping tool provided to case study participants for mapping their irrigation.

Phase 3: Each participant was invited to one of two workshops held in their community. During these workshops, the participants were asked to use MapChat interact with the map data recorded during Phases 1 and 2, and record their discussion regarding irrigation and water resource issues. Direct supervision/support was provided by the researchers during these workshops.



MapChat2 - Mozilla Firefox

Chat Dialog

Location: Entire Map
View data: All/Any Views
Display mode: ● Threaded ○ Sequential

Showing 27 of 34. Get more|all

David Smith: Feb 24 2:40a Reply
Can anyone mark the main take points within the extent of this map?
Showing 1 of 1 replies

Peter Jones: Feb 27 3:46a Reply
Hi David, I have drawn circles indicating the general locations, but I don't know them exactly. We'll have to get more people involved to complete this more accurately.

John Strike: Feb 24 3:14a Reply
Is everyone using analogue meters to measure water flow/usage? We're wondering about the new requirements for monitoring flows under the new RMA rights.
Showing 1 of 1 replies

Select/Draw Features

Select from: [All Visible Layers]
Select style:
Select mode: ● Add ○ Remove

Feature #87270 (remove)
selected - from *Farm Property Boundaries*
Feature #35757 (remove)
selected - from *Farm Property Boundaries*
Feature #51834 (remove)
selected - from *Farm Property Boundaries*

- *Length: 6475.8330726663*
- gid: 51834
- objectid: 133983

Phase 4: The final phase sought participation from the volunteers in a discussion using MapChat from home via the Internet. A two week period of time was allocated, during which the participants were asked to login to the website, and participate in an ongoing discussion.

11th GEOIDE Annual Scientific Conference
Vancouver, B.C., May 27-29, 2009

UNIVERSITY OF OTAGO NEW ZEALAND

NZ Landcare Trust
landcare action on the ground

GEOIDE

Researchers:
G. Brent Hall, University of Otago, NZ
Michael G. Leahy, Wilfrid Laurier University