NETWORKED DECISION MAKING

Setup and Administration Guide

Explains how to setup and test networked decision making using the standard configuration options.

Contents

1. Introduction

Welcome to version 2.0 of Networked Decision Making, a system to support asynchronous group decision making without or in conjunction with meetings and voting. This version builds on the basic version to make a more comprehensive platform aimed at organisations looking for an open decision making, discussion and action management platform. A focus for this version has also been to make the software simpler to use. Some aspects that were geared to a global platform for decision making have been scaled back. This version should be used by any organisation that requires a decision making and action framework. Our thinking is that this should be all organisations albeit some may not yet be sufficiently democratically minded to want to use this framework.

The overall background to the project is based on the global strategy book which has been published at http://www.scribd.com/doc/98216626/New-Global-Strategy and goes into some detail on the reasons for the project. In addition the high level requirements of a global decision making system have been summarised at http://netdecisionmaking.appspot.com/gdms/about/index and so they will not be repeated. Initial steps are likely to be made by voluntary organisations establishing and running these sort of systems on a smaller scale.

This document addresses the requirements for installation, configuration and on-going operation of the networked decision making software. The solution uses the excellent and award winning Web2py web framework and familiarity with this framework and python programming are advantageous. However it should not be necessary to have any prior knowledge to get up and running on the Google application engine (GAE).

Web2py can be deployed on many platforms and can utilise a very wide range of databases. If you want to move away from the app engine then you should refer to the relevant web2py documentation. The GAE platform has been chosen because it is scalable and free for limited use. As broadly speaking relational database platforms offer additional features if it works on GAE then it should work on any other platform.

This manual is slightly unusual in that the software consists of both a general decision making solution and some specifics that have been geared to its use as a platform for making decisions and forming a strategy to improve the world. We appreciate that this is a rather ambitious goal and that we are somewhat under-resourced to deliver this. However both demonstrating what may be possible and promoting the use of the software for decision making among smaller groups without a pre-established hierarchy is key. Many clubs, societies and small companies struggle with empowerment and decision-making as hierarchy, politics, and fear of confrontation frequently interfere. This solution aims to remove some of these issues by providing a degree of confidentiality and privacy for people to record their preferences. We see it as inevitable that social networking software will eventually move in this direction and we are rather surprised that it hasn't done so already.

2. What is an asynchronous decision making platform?

We appreciate this may not be a term that you are familiar with. However the idea is simply that decisions are made when sufficient people have agreed upon the answer and that agreement doesn't require to be reached at the same point in time as it typically does with meetings and elections. Arguably much of the success of email, messaging and the web is because it removed the synchronous requirement of the telephone. No such relaxation of requirements has been established for group decision making yet and this is seen as a significant hindrance to progress and an opportunity for the future. It's perhaps a little cynical to describe leadership as persuading and threatening people to accept it's OK that leaders are entitled to more than followers – however we do think sharing the burden of decision making may also help with improved resource sharing in future.

3. Installation

3.1.Python and Networkx installation

If you plan to use this software locally then you should install python and networkx. Python 2.7.8 is recommended as appropriate for your operating system. You can install networkx if you want to use and test locally and also selenium if you want to run the test scripts that have been developed. However a version of networkx has been created and is part of the package and this works with the google app engine. The manner of installing additional packages on Google App Engine is rather unclear – however networkx and some dependencies have been bundled into the site-packages folder of the web2py installation and these can be used as long as you are not concerned about conflicts with other packages running on the same web2py environment.

3.2.Networked Decision Making Installation

Installation of the software requires a minimum of two steps:

1 Install the software locally (and ensure operational)
2 Deploy the software to Google App Engine

Installing the software locally should consist of extracting the package which you can download from http://netdecisionmaking.appspot.com/gdms/about/download and then following the instructions for installing a packed application ie use the upload and install a packed application which is available at http://127.0.0.1:8000/admin/default/site with a default web2py installation. It is recommended that you get the solution working locally using web2py and the SQLite database prior to testing on the Google app engine. The deployment process once it works locally should be simple. Please also note you will need to install the source code version of web2py if you want to deploy on the Google app engine as this is the only mechanism supported. For local testing the executable web2py can be used.

The instructions to install web2py are available at web2py.com and consist of downloading and extracting the files. Following this you can just download the package from here and install as a package using the web2py admin application. It is recommended that you use the latest production release of Web2py (currently 2.9.11) for deploying the software as there are some changes in this release regarding the web technologies used and earlier versions may have some issues.

Downloading the package:

Go to http://netdecisionmaking.appspot.com/gdms/about/download and download

Installing the package

Open up web2py and choose a name for the package the default will be gdms but it should work fine with any chosen name.

3.3.Advanced installation points

While the above setup will get the basic system working there are 3 issues to consider if you want to deploy.

1 The index.yaml file
2 Email login setup
3 Remote authentication (Janrain)

Index.yaml

Unpacking the application or cloning from the project home page at code.google.com/p/global-decision-making-system/ will get you the latest index.yaml file for the software which is required to run immediately on the google app engine. However this file is in the WRONG folder and would need to be moved to the root web2py folder before use. However as you may have other applications using a common index.yaml the application does not directly copy it there. You should either do this manually overwriting what you have if it's not important or merge with any other applications sharing the same web2py installation.

E-mail

To use the email features of the application you need to create a simple text file named emaillogin.key with the format username:password but obviously with the valid values for the email account you want to use and other mail settings would need to be configured if you don't want to use Google for the mail server.

Janrain

You would need to setup a key file called janrain.key as described in the web2py manual after you have registered with Janrain to get the appropriate details.

Networkx

Installation of networkx on GAE was less than tidy – current approach was basically to copy this and everything else required into the web2py site packages directory. A separate download of this will be made available – however it seems there ought to be a better method of doing this. Networkx also relies on NumPy so this should be enabled in the app.yaml file.

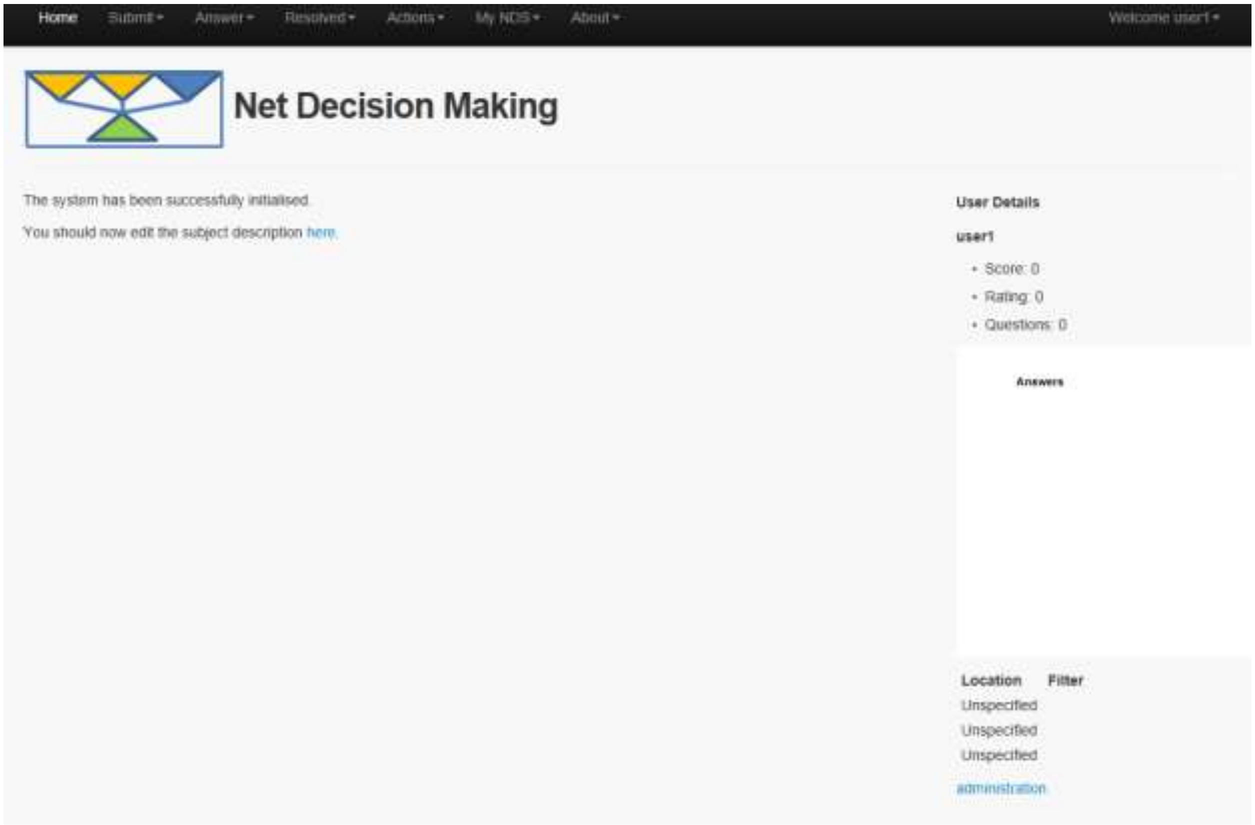
4. Background

It is important to note that deployment of the software to the Google app engine does NOT deploy any data. This allows you to test setups and make adjustments prior to deployment. When the software runs for the first time minimal data is setup to allow the application to run. However there are some further setup activities that should be considered.

The package does not have any data included and following initial deployment the first user will be prompted to register and automatically becomes an administrator of the system. This allows running of the various admin programs and allows you to setup the data you want for your version of the software. There are generally also maintenance options available to supporting editing the default data or setting up from scratch with your own values.

5. Configuration and setup

On initially starting the application you will need to register as user using the web2py registration and the first person to register will automatically be setup as an administrator of the system. The trigger for the registration process to trigger is that there is no record for the description and scope of the system in the website parameters table. Once completed you are taken to a page with a link to edit these values and explain the scope of the system that you are setting up:



The system has been successfully initialised.

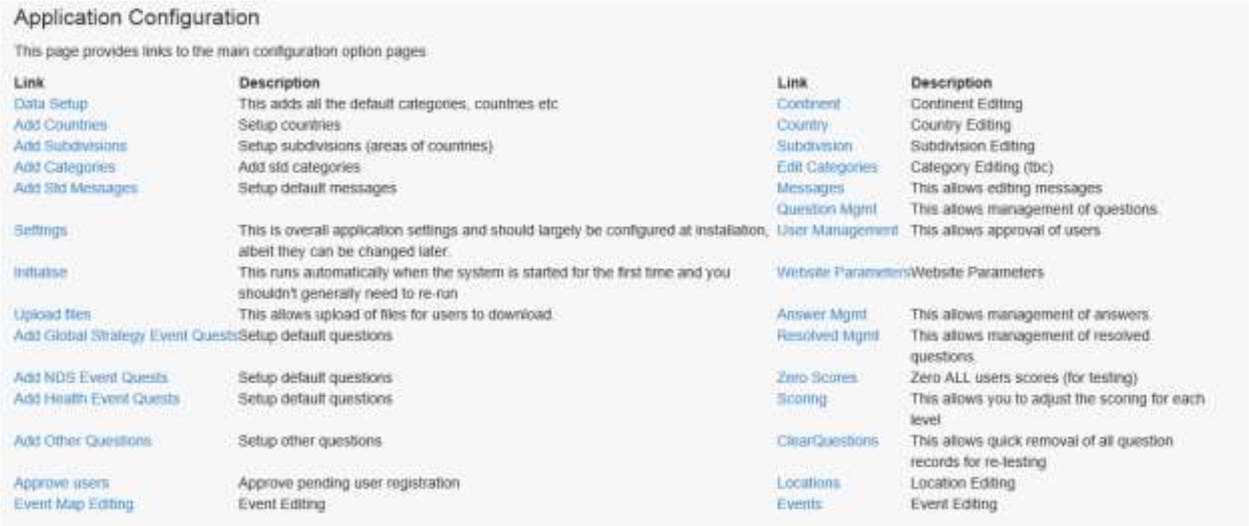
You should now edit the subject description here.

- Score: 0
- Rating: 0
- Questions: 0

Location	Filter
Unspecified	
Unspecified	
Unspecified	

5.1. Subject of the networked decision making system

You should add a description of the intended scope and purpose of this particular instance of the networked decision making software is and who the intended audience and use is. After that is completed you should click on the administration link at the bottom right and this takes you to the main configuration and setup options screen which looks something like this:



Application Configuration

This page provides links to the main configuration option pages

Link	Description	Link	Description
Data Setup	This adds all the default categories, countries etc	Continent	Continent Editing
Add Countries	Setup countries	Country	Country Editing
Add Subdivisions	Setup subdivisions (areas of countries)	Subdivision	Subdivision Editing
Add Categories	Add std categories	Edit Categories	Category Editing (tbc)
Add Std Messages	Setup default messages	Messages	This allows editing messages
		Question Mgmt	This allows management of questions.
Settings	This is overall application settings and should largely be configured at installation, albeit they can be changed later.	User Management	This allows approval of users
Initialise	This runs automatically when the system is started for the first time and you shouldn't generally need to re-run	Website Parameters	Website Parameters
Upload files	This allows upload of files for users to download.	Answer Mgmt	This allows management of answers.
Add Global Strategy Event Quests	Setup default questions	Resolved Mgmt	This allows management of resolved questions.
Add NDS Event Quests	Setup default questions	Zero Scores	Zero ALL users scores (for testing)
Add Health Event Quests	Setup default questions	Scoring	This allows you to adjust the scoring for each level
Add Other Questions	Setup other questions	ClearQuestions	This allows quick removal of all question records for re-testing
Approve users	Approve pending user registration	Locations	Location Editing
Event Map Editing	Event Editing	Events	Event Editing

The configuration and setup options on this screen are on the left, whereas some screens to support on-going data editing are available on the right. For a standard demonstration system you should complete the following steps

5.2.Main Setup options

5.2.1. Data Setup

Data setup is recommended to be run for all implementations as this is minimal operational data.

5.2.2. Add Countries

This step is optional and if not carried out then all questions will be categorised with a country of unspecified. This is recommended for local setups where geographic information is irrelevant.

5.2.3. Add Subdivisions

This currently sets up US States and Canadian provinces only. However it should be readily possible to extend to other countries. The view is that local decision making is important and so this should be supported, albeit use will probably not be terribly effective until a system obtains critical mass of users.

5.2.4. Add Categories

This adds the standard categories. In most circumstances for local use these would require to be tailored to the subject that the system is designed to cover.

5.2.5. Add std messages

These are probably not required for most systems. The purpose of putting them here is to try and both promote the system and to encourage people to feel more empowered and less constrained than they may do at present. However for most versions these probably wouldn't be required in their present form.

5.2.6. Add Event Questions

These are some sample questions and actions which are designed to demonstrate the sort of questions that we may need to try and agree on in order to make faster progress towards a global strategy. However the key point of the system is to allow all users to generate questions for resolution so this is just some possibilities to get started. This has now been split into 3 event related groups of questions and some other questions. These are really for demonstration purposes only – it is really envisaged that users will add and link questions themselves.

5.2.7. User Administration

The system requires user authentication in order to operate. It can be configured to either use 3rd party authentication systems such as google or facebook or to use web2py for authentication. In either case you need to decide if users need to have their email addresses verified or actual administrator approval prior to login. These are

5.2.8. Categories

Use of categories to classify questions is optional however they can help with supporting specialisation of questions and allowing different people to focus on different questions. The category option is always present however the administrator can limit the number of subjects or just leave with the default of unspecified if there is no desire to manage questions by subject.

5.2.9. Geographic Scope and localisation

Localisation of questions by continent, country and sub-division is always present however there is no need to use it. Indeed if the administrator doesn't setup the continents, countries and subdivisions then all questions will be forced into an unspecified category. This would be the recommended setup for small groups or limited geographies.

5.2.10. Default e-mail messages

There is a standard message to explain the aim of the system to key people that might be significantly implemented by it. However this can be sent to anyone as we all need to help with leading the world to a better future.

6. Configuration Settings

Currently configuration options are mainly related to setting up the authentication and verification options that you want to use. There are 3 main settings

1 Require Verification
2 Require Approval

The first two are standard web2py options which are fully explained in the web2py documentation. With verification turned on users will require to verify their email before they can sign on again following registration. With approval required the users would need to be approved by an administrator prior to being given access. This would be appropriate if the system was to be used with a restricted audience eg members of a club or a particular special interest group that isn't open to the public.

7. On-going operation

The basic approach is that the system should operate in a pretty much maintenance free manner once setup. In particular unsuitable questions will be rejected if a majority of respondents at any level vote for this to happen and comments will be removed as soon as 3 people deem them inappropriate and there is no appeal's process involved in either case. Clearly this approach could be changed if you wanted to monitor activity however this is not considered particularly necessary.

The admin page does have a number of links to permit editing of most of the key tables that are present – however it is not envisaged that these would require to be used on a regular basis. Other database administration options are available via standard web2py methods and also using the datastore administration features of the relevant database that is connected eg the appengine control console.

The right hand side of the admin page is basically the editing features that have been setup and the left hand side is for loading the initial setups that you might require.

8. Testing

The software comes complete with a suite of tests that have been developed based on the selenium test framework for automated browser based testing using Python. The test suite is stored in the fts folder and should be run from the main package folder using a command line prompt.

Eg python .\fts\functional_tests.py

This command would run the whole test suite – however different phases of testing can also be run independently, but there are some dependencies in later tests particularly on the basic data and test users having been setup for testing. There are still some minor issues with testing on the google app engine due to the issue of eventual consistency of queries and the fact that updating data and immediately requerying may NOT return the updated or inserted data. This is still being worked through but the tests should all pass when running locally using SQLite.

A summary of the test phases and what they do is noted below:

Phase	Description	Purpose and dependencies
0	This is basic setup of user 1 and is necessary for subsequent testing	
1	This is setup of users 2,3 and 4 plus	
a-o Unnumbered	This is basic testing of static pages etc	
2	This involves adding further questions	
3	This covers challenging questions and should end up with resolution of a challenged question	
4	This is a round of std testing with 3 users and all questions should get the first answer and end up resolved	
5	This is a further round of std testing of questions but they are generally reclassified by scope and category as part of answering	
6	This is a round of testing of actions	
7	This covers viewing questions and actions by category and location	
8	This covers creating events and linking questions to events	

9. Questions and Issues

Any questions, issues or suggestions for improvement of the software can either be logged on the development site at https://code.google.com/p/global-decision-making-system/ or via email to Russ at newglobalstrategy@gmail.com

10. Other Issues

Once you have a working system obviously one of the key things is to figure out how it fits in with your existing decision making structures. This is not a software problem and is something each group will have to work out for themselves. Doubtless however decisions made here may well create some requirements for enhancements to help with tracking and recording. We are happy to look to develop any sensible suggestions here and the thinking is that proposals could be agreed via the networked decision making system that it could then evolve itself. The initial demand for this is to get some quality feedback and suggestions on how to take this forward.