# FEARLUS (Framework for the Evaluation and Assessment of Regional Land Use Scenarios)

Nick Gotts, Gary Polhill & Alistair Law, Macaulay Institute, Craigiebuckler, Aberdeen AB15 8QH United Kingdom

http://www.macaulay.ac.uk/fearlus

Model Objectives: FEARLUS is a research project aimed at building agent-based models of land use change. Our approach is to start with simple, abstract models, building in extra complexity on the basis of a thorough understanding of the simpler models. This requires us to code the models in as flexible a way as possible, plugging new modules in to the model as required. This flexibility is provided by languages such as Objective C and Java.

Software: The current FEARLUS model (0-5) is written in Objective C, though we have plans to build a new modelling framework (1-0) using Java.

The Swarm programming libraries fit well with the kind of work we want to do - giving us the freedom to code the agent behaviour and environment functionality the way we want, whilst providing a set of tools to do things we have less expertise in or time to do properly:

GUI visualisation - Creating graphs and rasters to display data during the simulation is easy with Swarm. This was one of the key motivations for using Swarm

Random number generation - The Swarm libraries provide a number of different up-to-date random number generating algorithms. This saves us from keeping up with the literature in this area, as well as giving us the potential to check for any biasing effects of random number generation in our results.

"Best practice programming" - We learned a great deal from looking at the example Swarm applications. These showed us how to structure an agent-based simulation, and in particular, emphasised the importance of separating the observer from the model (a key benefit of using computer simulations over real-world studies!)

Data structures - Swarm provides standard data structures such as lists, arrays and hashes, as well as spatial and cellular automaton data structures.

Scheduling - Swarm scheduling integrates the observer and model schedules. Using Swarm schedules also forces the programmer to structure the simulation in a principled fashion.

Memory management - The use of Zones makes memory management in Objective C much less of a headache.