

A MULTI-MODEL DOCKING EXPERIMENT OF DYNAMIC SOCIAL NETWORK SIMULATIONS

Jin Xu, Yongqin Gao, Jeff Goett, Gregory Madey
**Department of Computer Science and
Engineering**
University of Notre Dame
Notre Dame, IN 46556

Supported by NSF, CISE/IIS-Digital Science and Technology
October 2, 2003

OUTLINE

- INTRODUCTION
- SOCIAL NETWORK MODEL
- OPEN SOURCE SOFTWARE (OSS) NETWORK
- DOCKING
- RESULTS
- CONCLUSIONS & FUTURE WORK

INTRODUCTION

- Three ways of Validation
 - Comparison with real phenomenon
 - Comparison with mathematical models
 - Docking with other simulations
- Docking
 - Verify simulation correctness
 - Discover pros & cons of toolkits

OSS DOCKING EXPERIMENT

- **Four** Models of OSS Developer Social Network
 - Random graphs
 - Preferential attachment
 - Preferential attachment with constant fitness
 - Preferential attachment with dynamic fitness
- Agent-based Simulation
 - **Swarm**
 - **Repast**

SOCIAL NETWORK MODEL

- Graph Representation
 - **Node/vertex** – Social Agent
 - **Edge/link** – Relationship
 - **Index/degree** - The number of edges connected to a node
- ER (random) Graph
 - Edges attached in a random process
 - No power law distribution

SOCIAL NETWORK MODEL(Cont.)

- Watts-Strogatz (WS) Model
 - Include some random reattachment
 - No power law distribution
- Barabasi-Albert (BA) Model with Preferential Attachment
 - Addition of preferential attachment
 - Power law distribution
- BA Model with Constant Fitness
 - Addition of random fitness
- BA Model with Dynamic Fitness

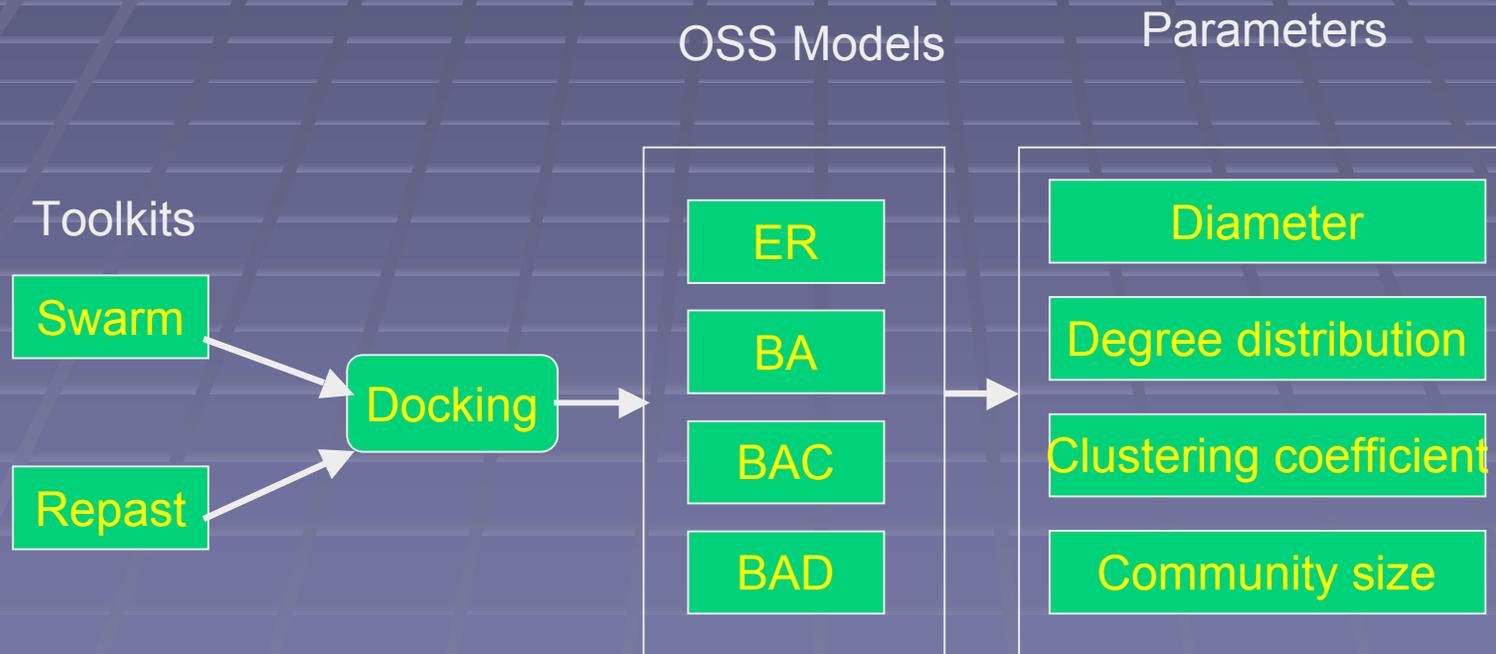
OSS NETWORK

- A Classic Example of a Dynamic Social Network
- Two Entities: developer, project
- Graph Representation
 - Node – developers
 - Edge – two developers are participating in the same project
- Activities
 - Create projects
 - Join projects
 - Abandon projects
 - Continue with current projects

OSS MODEL

- Agent: **developer**
- Each Time Interval:
 - Certain number developers generated
 - New developers: create or join
 - Old developers: create, join, abandon, idle
 - Update preference for preferential models

DOCKING PROCESS



SWARM SIMULATION

- ModelSwarm
 - Creates developers
 - Controls the activities of developers in the model
 - Generate a schedule
- ObserverSwarm
 - Collects information and draws graphs
- main
- Developer (agent)
 - Properties: ID, degree, participated projects
 - Methods: daily actions

REPAST SIMULATON

- Model
 - Creates and controls the activities of developers
 - Collects information and draws graphs
 - Network display
 - Movie
 - Snapshot
- Developer (agent)
- Project
- Edge

DOCKING PROCEDURE

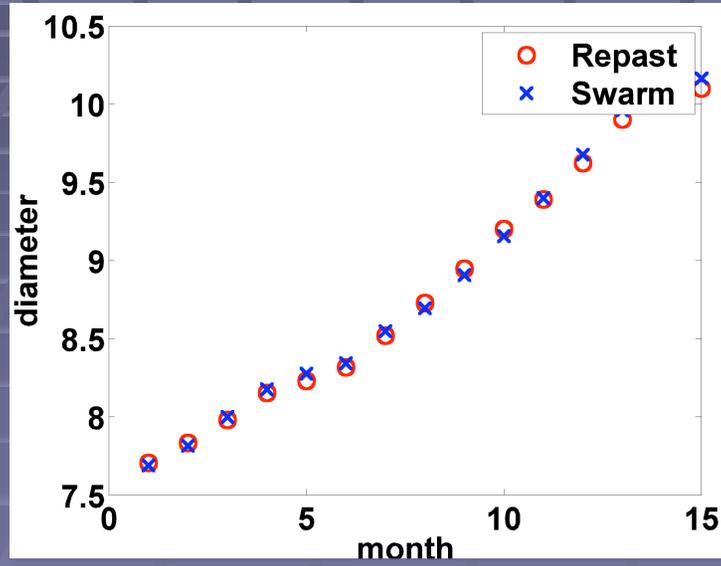
- Process: comparisons of parameters corresponding models.
- Findings:
 - Different Random Generators
 - Databases creation errors in the original version
 - Different starting time of schedulers

DOCKING PARAMETERS

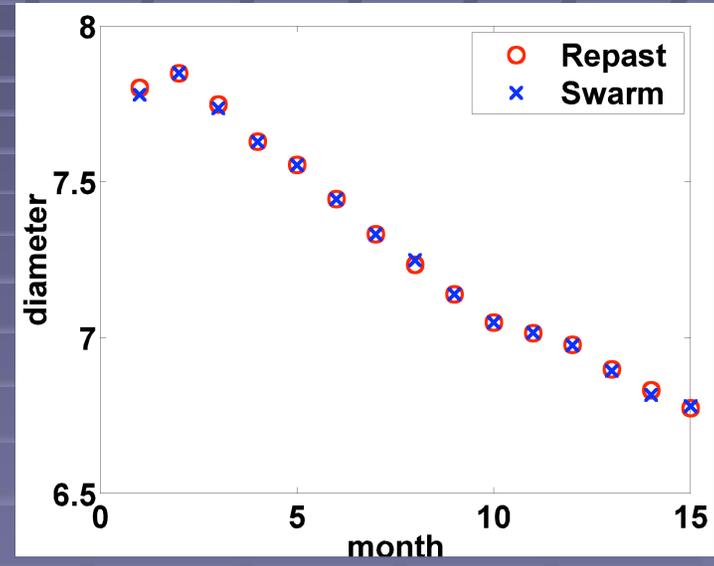
- Diameter
 - Average length of shortest paths between all pairs of vertices
- Degree distribution
 - The distribution of degrees throughout a network
- Clustering coefficient (CC)
 - CC_i : Fraction representing the number of links actually present relative to the total possible number of links among the vertices in its neighborhood.
 - CC: average of all CC_i in a network
- Community size

DIAMETER

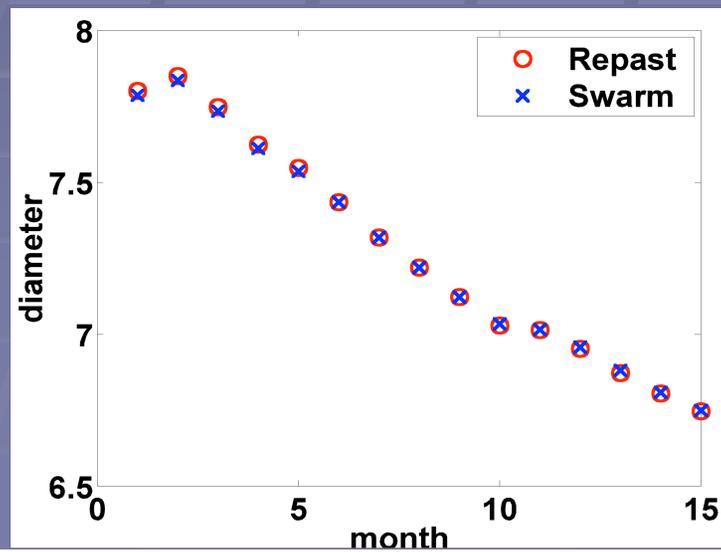
ER



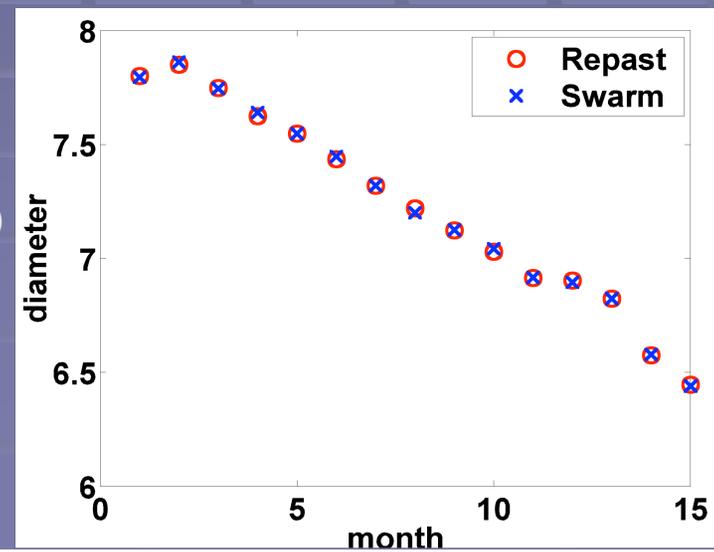
BA



BAC

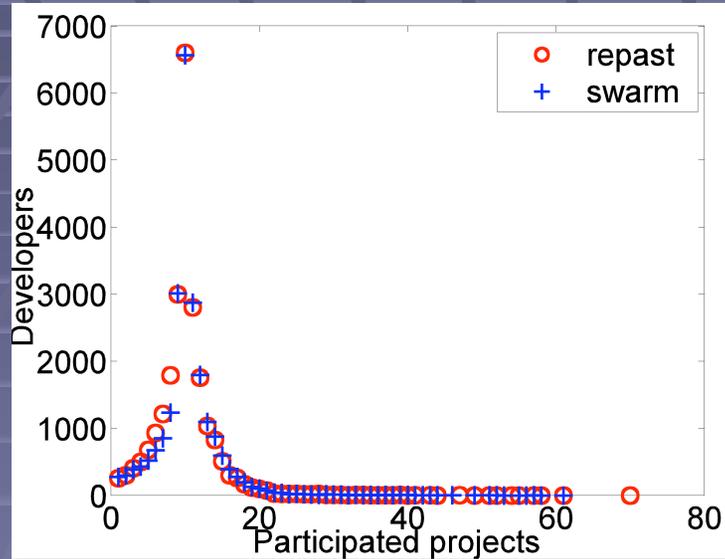


BAD

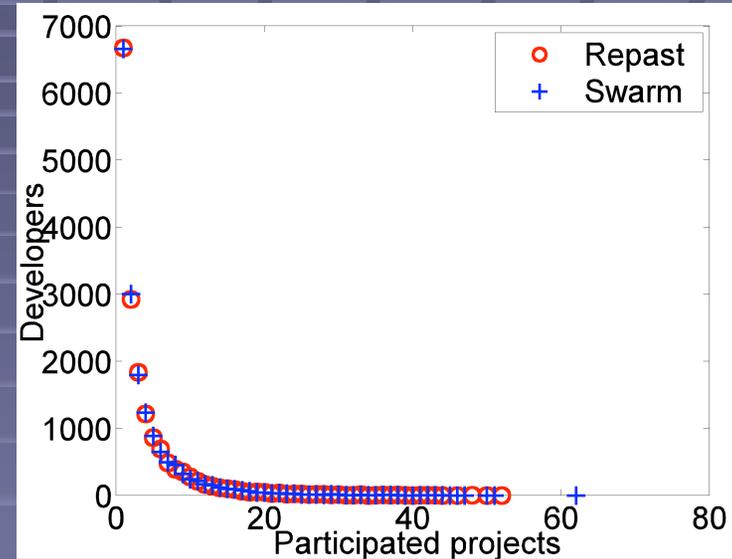


DEGREE DISTRIBUTION

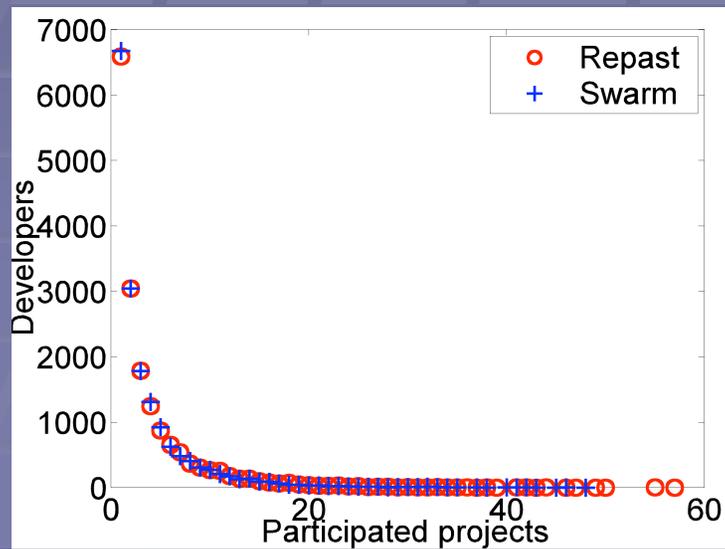
ER



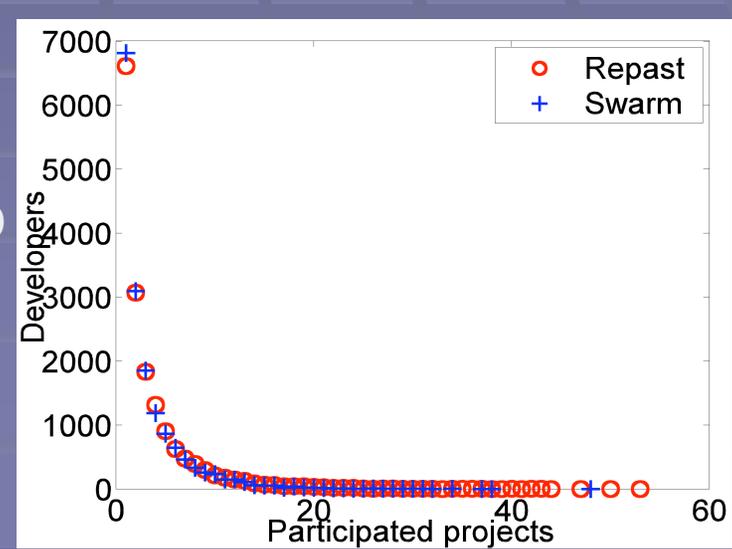
BA



BAC

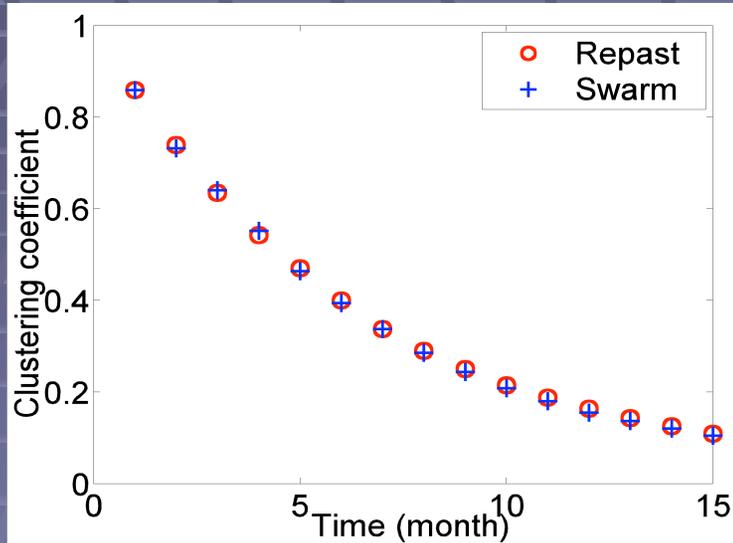


BAD

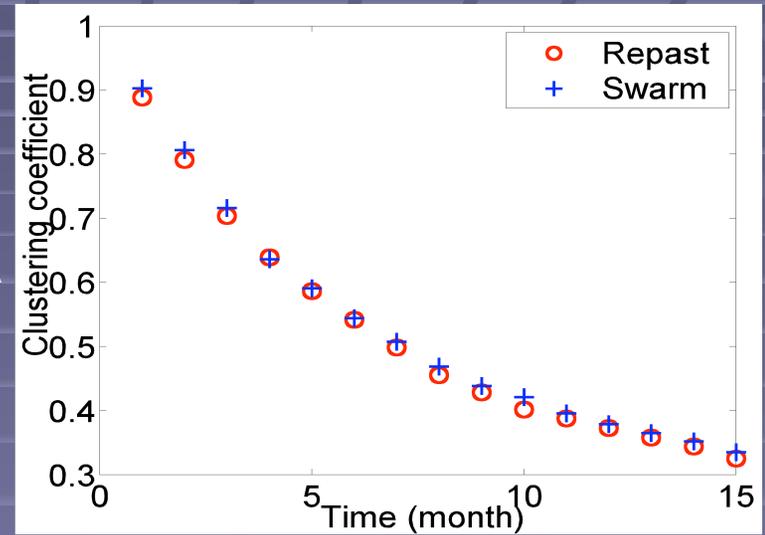


CLUSTERING COEFFICIENT

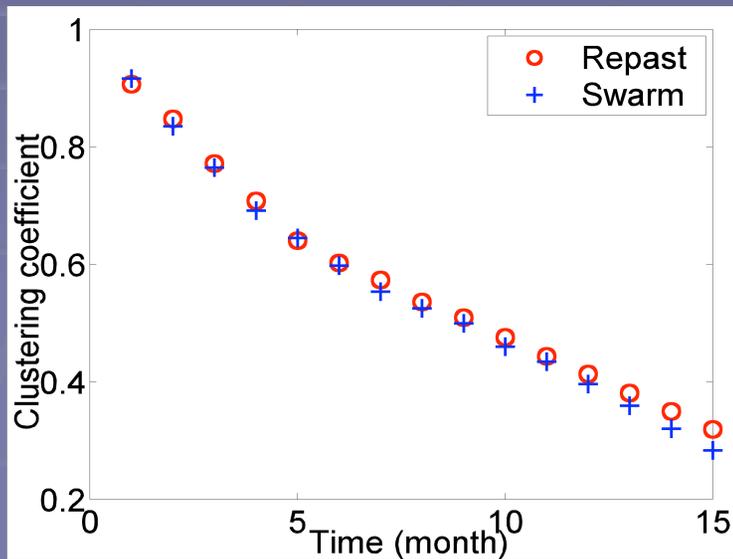
ER



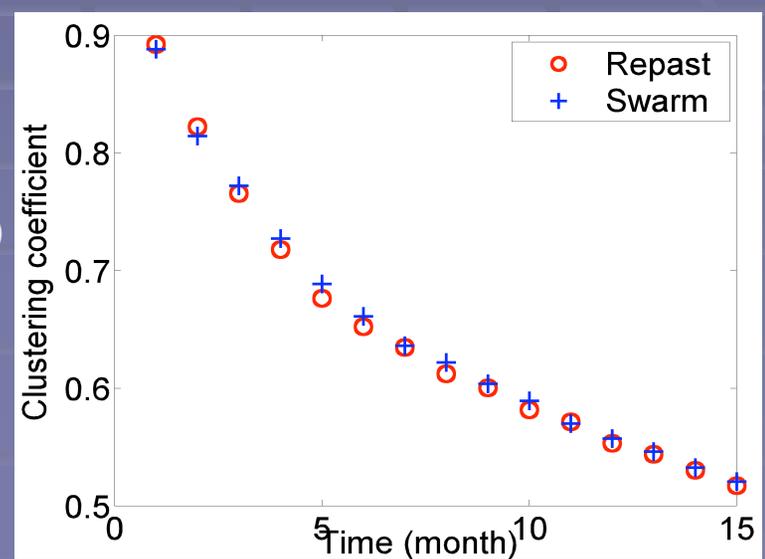
BA



BAC

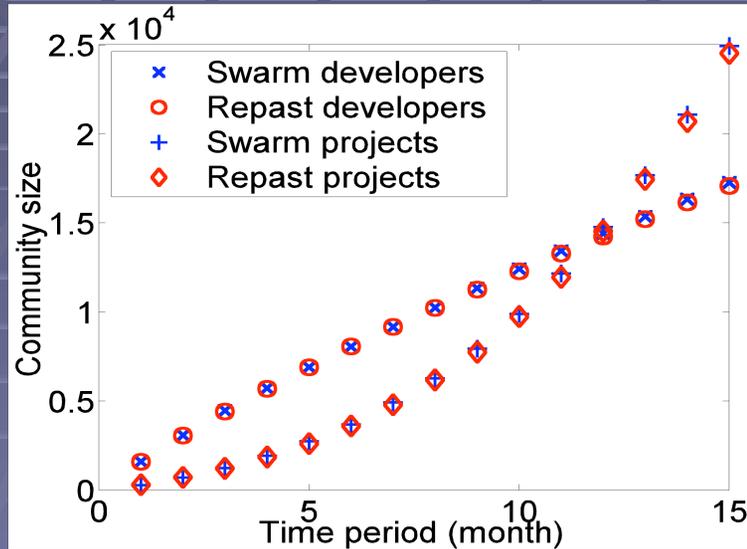


BAD

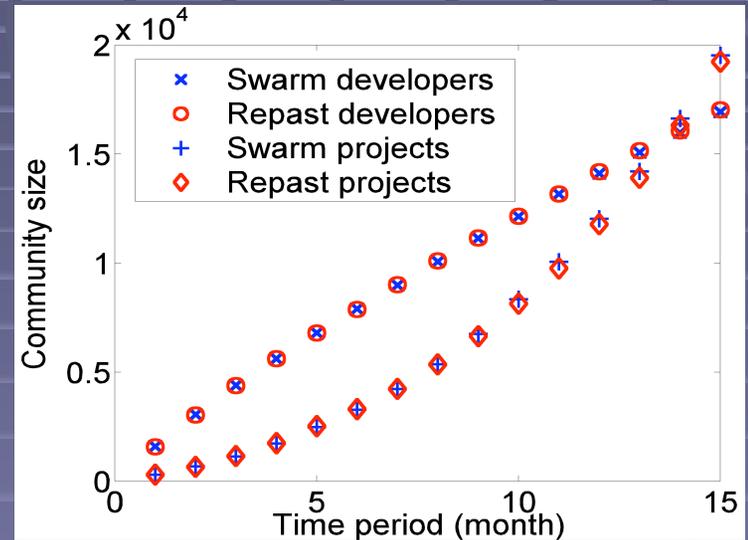


COMMUNITY SIZE DEVELOPMENT

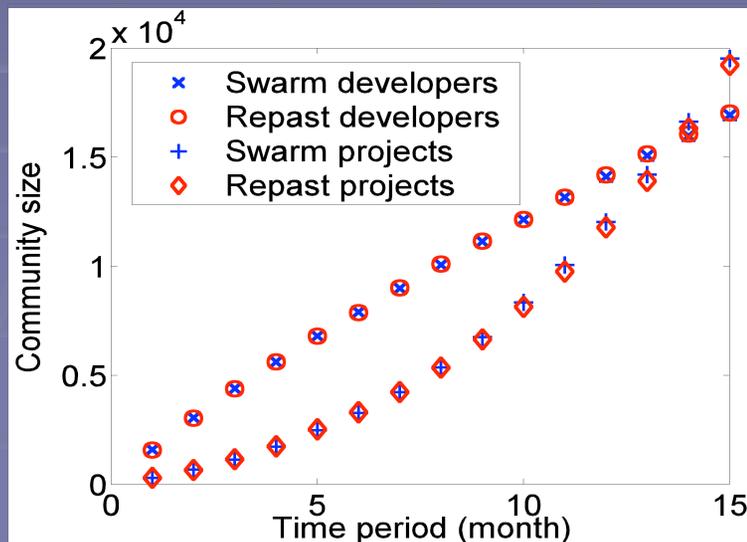
ER



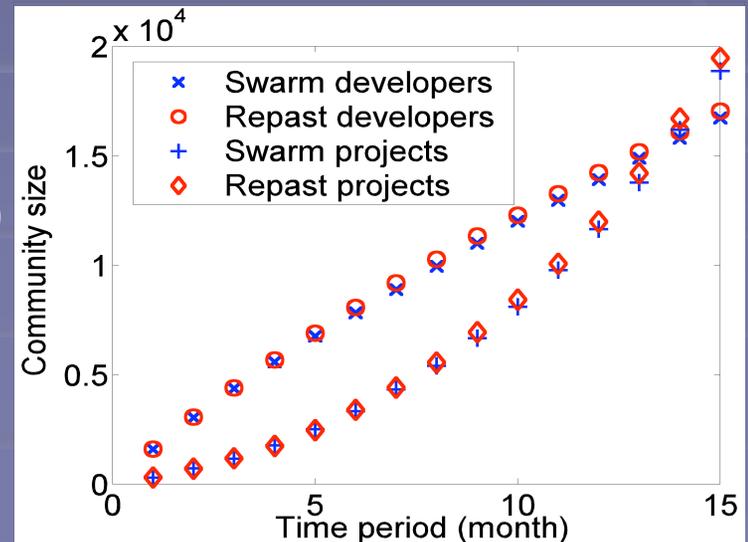
BA



BAC

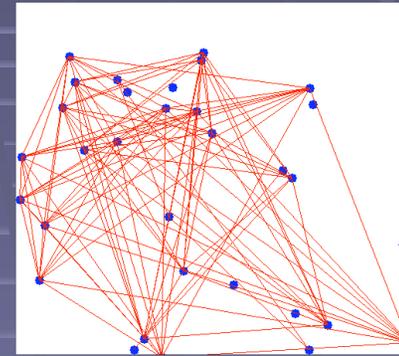


BAD

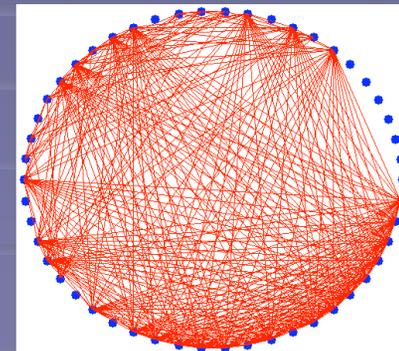


CONCLUSION

- Same Results for Both Simulations
- Better Performance of Repast
- Better Display Provided by Repast
 - **Network display**



Random Layout



Circular Layout

FUTURE WORK

- Many runs instead of one
- Statistical analysis
- More network parameters
 - Average degree
 - Cluster size distribution
 - Fitness and life cycle