Verification and Partial Validation of the Sim-I-Space Simulation Model

Max Boisot
Universitat Oberta de Catalunya, Barcelona
Sol Snider Entrepreneurial Research Center, The Wharton School, University of Pennsylvania, Philadelphia.
boisot@attglobal.net

Ian MacMillan
Sol Snider Entrepreneurial Research Center,
The Wharton School, University of Pennsylvania, Philadelphia.

Kyeong Seok Han
Sol Snider Entrepreneurial Research Center,
The Wharton School, University of Pennsylvania, Philadelphia
Soongsil University, Seoul, 156-743, Korea.
kshan@wharton.upenn.edu

Casey Tan
Singapore Airlines.
Formerly of Sol Snider Entrepreneurial Research Center, The Wharton School, University of Pennsylvania, Philadelphia

Si Hyung Eun
Sol Snider Entrepreneurial Research Center, The Wharton School, University of Pennsylvania, Philadelphia.

Acknowledgement: The authors would like to thank the Wharton – SMU Research Program of Singapore Management University for their financial support in carrying out this research

1. **Introduction**

This paper offers a brief description on how the Sim-I-Space was verified and validated. Sim-I-Space is set up to run on a Linux-based platform via a "Swarm terminal", a Windows based Linux emulator, in three major stages as shown in Figure 1. The first stage initializes the variables and sets the input parameters. We have a "variables.h" file that defines 64 variables, and into this file we input parameters that reflect different real-world situations. The second stage runs the simulation model. Ten files represent the objects, and the main file (main.m) calls the other nine files, depending on the simulation procedures. The third stage generates output statements for printing results. The second and third stages run concurrently while the simulation program is running on the "Swarm-terminal".

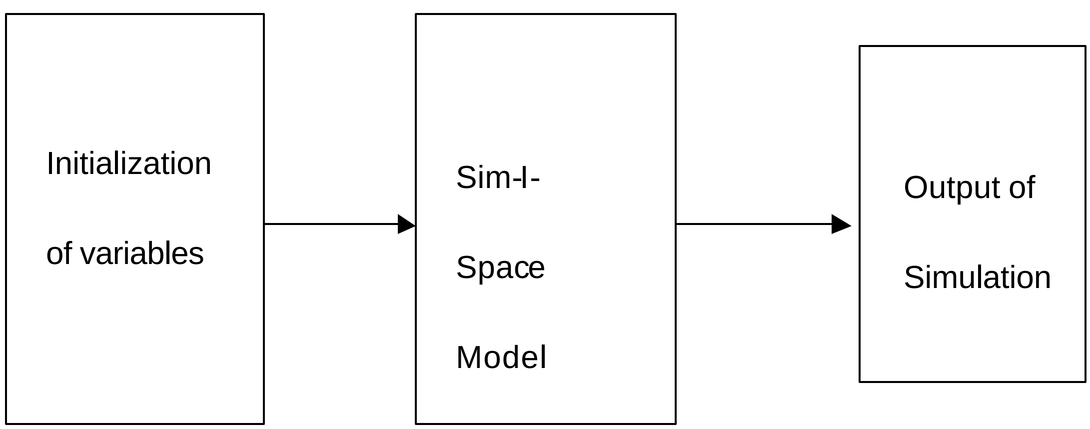


Figure 1 The Structure of Sim-I-Space Simulation System

1.1. Initializing Sim-I-Space variables.

In this stage users must set all of the variables that get the simulation going. The main challenge here is to set the parameters that reflect the real-world situation being simulated with some degree of realism. In addition, users can set up the number of periods required, where the time unit is usually a quarter - i.e. one period is a quarter of the year. Forty periods, therefore, covers 10 years. We used 80 periods (i.e. 20 years) for the simulation runs because over that period the variables tend to settle down to more realistically stable values.

Each agent enters the simulation with a randomly assigned initial endowment – ie, a number of randomly generated nodes and links representing discrete knowledge assets, some financial funds (used to cover the costs of activities such as meeting other agents, investment costs and so on), and some experience funds (used to cover the costs of generating and enhancing their knowledge assets).

1.2. Running the simulation.

In this stage, 10 object files work together based on the 12 modules listed below:

1. Asset properties:
2. I-Space Matrix and I-Space Locations
3. Linkage Probability Matrix.
4. Agent properties
5. Movement and/or Creation of assets
6. The Creation of new assets
7. Agent meetings
8. Presentation and inspection
9. Meeting costs
10. Meeting transactions
11. Obsolescence decay
12. Diffusion decay

1.3. Ending the simulation and outputting results At the end of each time period, the numbers of Nodes and Links are cumulated and the system enters a clean-up phase.

Nodes and Links are first diffused through the action of the diffusion decay function and then partially obsolesced through the action of the obsolescence decay function. These processes are repeated until the last period of the simulation run.

The processes just described are flowcharted in Figure 2 below:

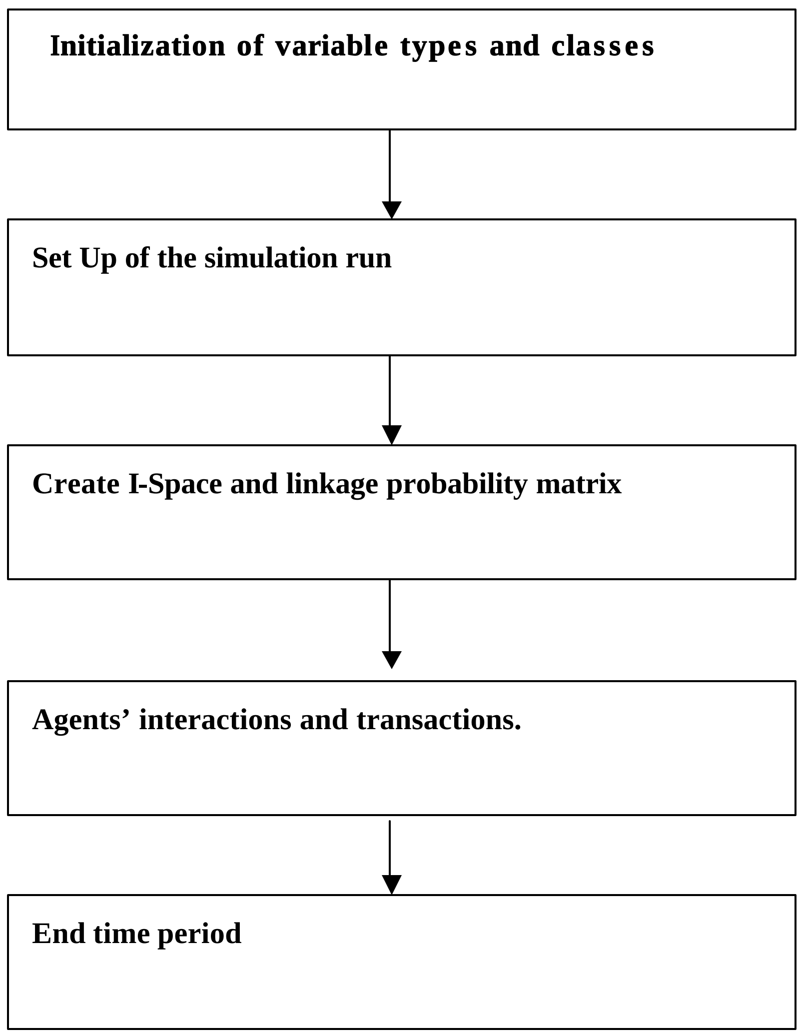


Figure 2: The Sim-I-Space Processes

2. Verification of the program code.

We developed Sim-I-Space through a Protocol Development Methodology as indicated in Figure 3 and Table 1:

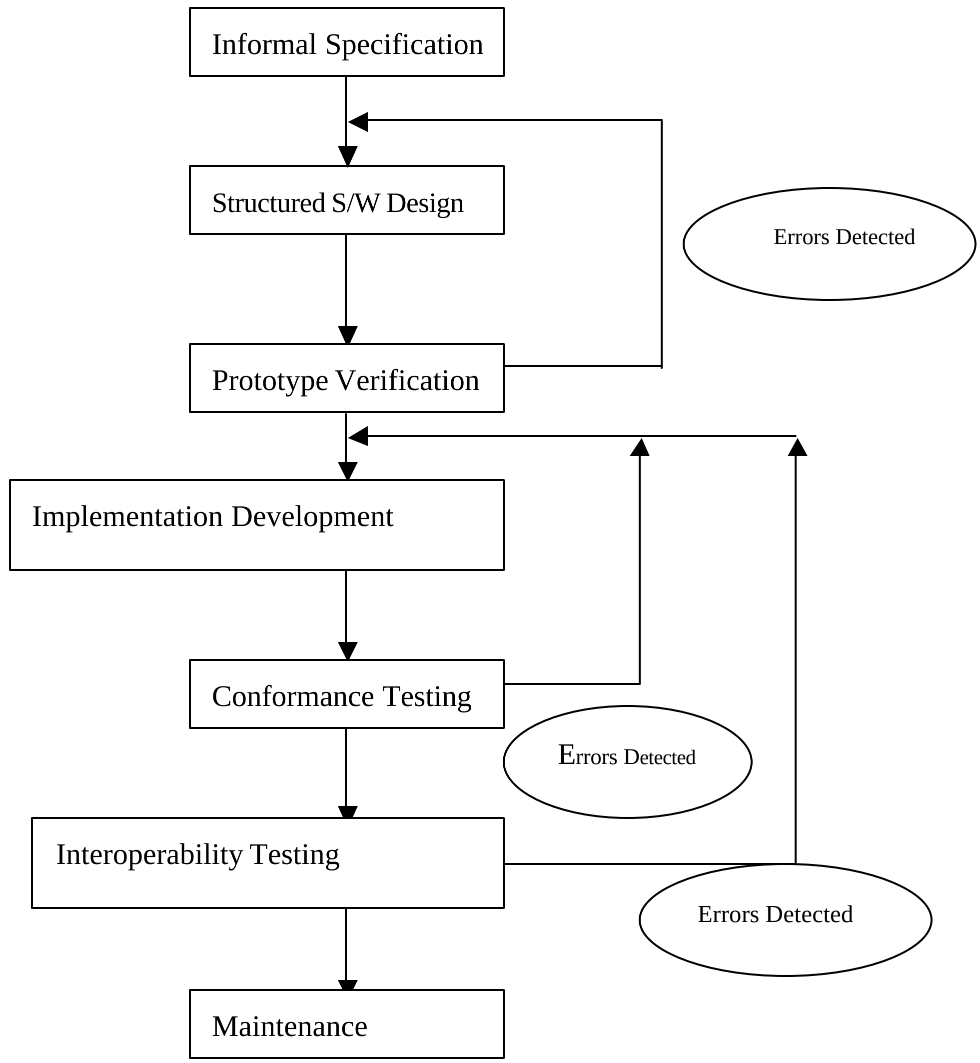


Fig2 Prototype Development Methodology

Informal Specification: The model specifications were based on the theory of the I-Space (Boisot, 1998).

Structured Software Design: The specifications used a structured software design employing HIPO (Hierarchy-Input-Process-Output) diagrams and structured flowcharts[1] which allowed the specifications to be analyzed for program verification.

Prototype Verification: A prototype was developed first. A full version of the Sim-I-Space simulation software was then developed and verified.

Implementation Development: The program code of the simulation software was developed according to the specifications in the structured software design methodologies.

Conformance testing: The program was hand-checked step by step for conformance. Through systematic testing the protocol was fed a selected set of test inputs, the outputs generated and then observed and checked against specifications. Errors were removed and the conformance test repeated.

Interoperability Testing: The program was run repeatedly at a selected set of parameter settings, and checked for consistency of output. Sources of errors were identified and corrected, then the simulations rerun till consistent outputs were obtained. Then parameter settings were cha nged and the process repeated until consistent results were obtained for the parameter ranges specified in Appendix 2.

3. Partial Validation of Sim-I-Space: Some Results

In what follows we describe a set of Sim-I-Space validation runs in which we explor e whether the effects of diffusion decay and obsolescence decay that occur in the simulation conform with expectations. Note that in this paper we are looking to validate – which means we are concerned with whether we results that we expect under varying conditions of diffusion and obsolescence decay. We are not here concerned with combined effects – this is the topic of a later paper**.**

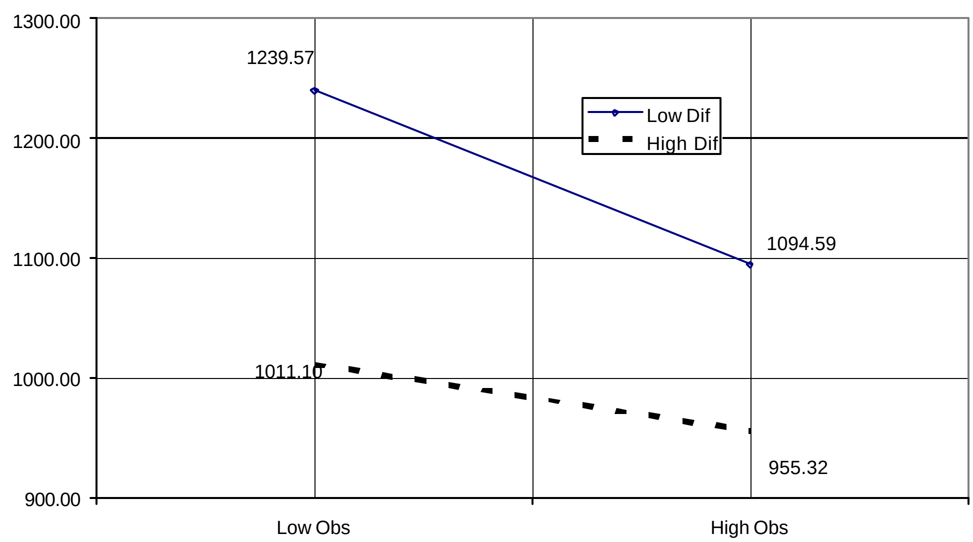
[1] Richard Fairley, Software Engineering Concepts, McGraw-Hill, 1985, pp.137-191

To validate the model we set parameters and held them steady for all variables except diffusion and obsolescence decay. Appendix 2 lists the parameter settings for the runs.
We then set each simulation to run for 80 periods. Four combinations of decay were simulated – high and low diffusion with high and low obsolescence decay. Each combination was run 200 times, to give us sample sizes of 200 for each "corner" (high/low diffusion vs. high/low obsolescence)

Below we first present a number of graphic illustrations of diffusion decay and obsolescence decay at work on selected outcome variables and comments on the validity of these outcomes. Appendix 1 presents tabular summaries of model runs. These tables replicate the graphs, but in the tables we indicate the manifold places where the differences we see in the graphs are statistically significant.

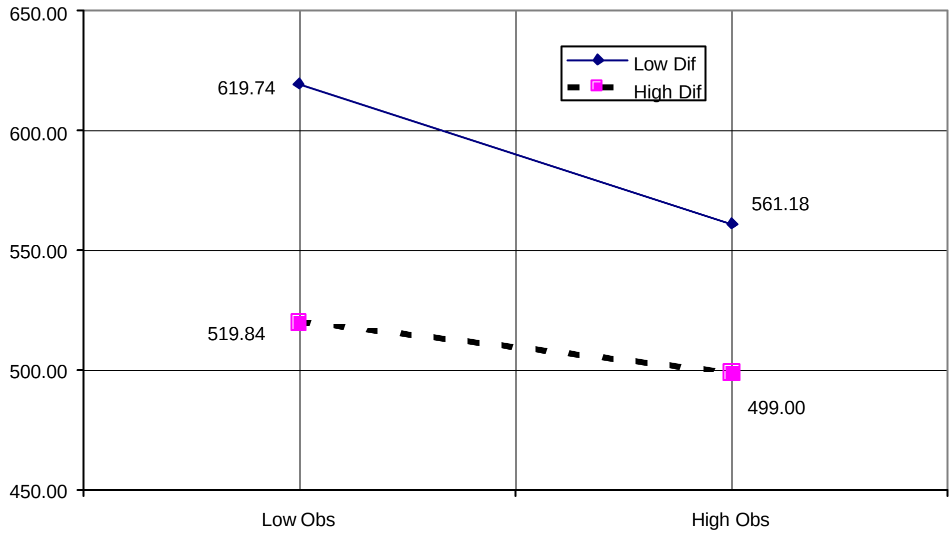
RESULTS

1) **Average number of nodes per agent at the end of 80 periods**



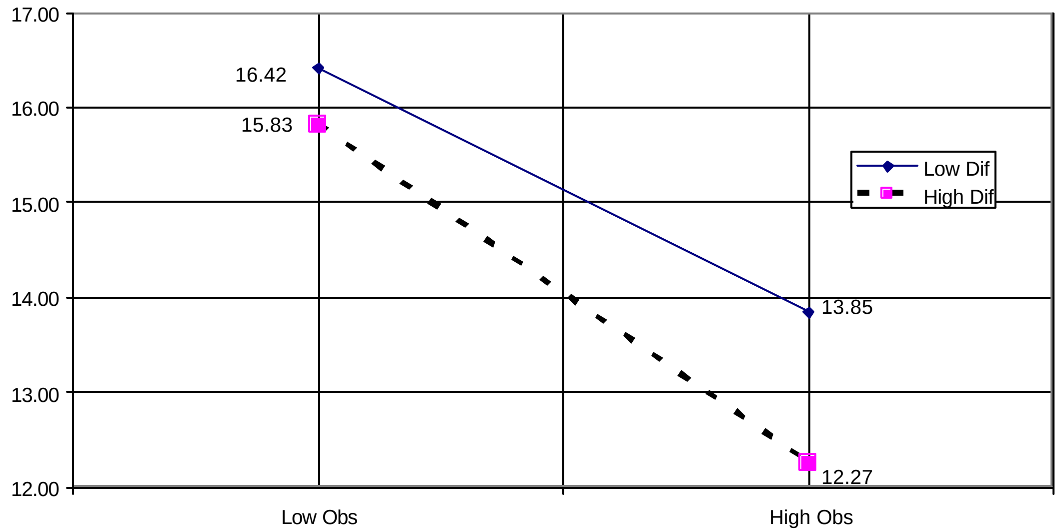
Increasing diffusion and obsolescence rates will have the effect of eroding the knowledge base of agents over time. All other parameters being equal, at the end of the simulation we would therefore expectthat agents subject to high obsolescence would have fewer surviving knowledge **nodes** than those experiencing low obsolescence. The same applies to agents experiencing high compared to low diffusion. The graph above shows that this is the case, showing that the simulation was performing as expected for our settings.

2. **<u>Average number of links per agent at the end of 80 periods</u>**



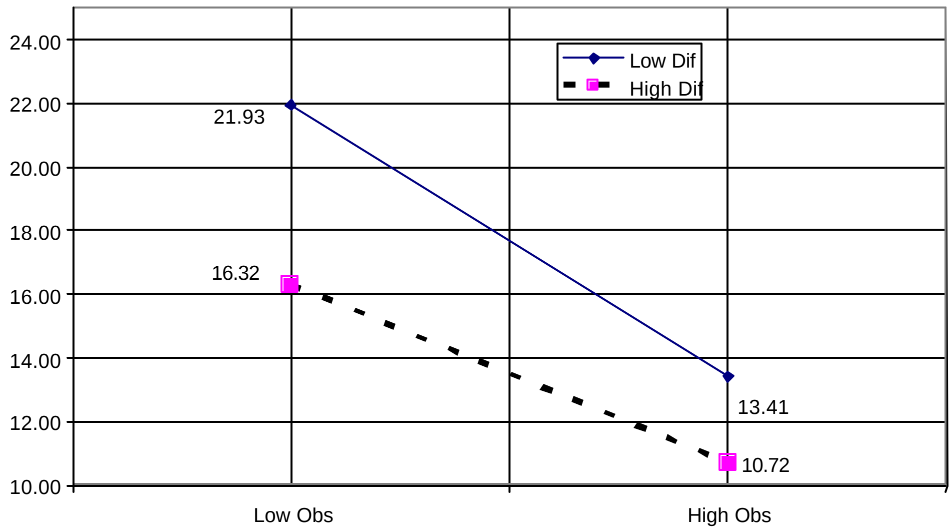
As in the case of knowledge nodes, increasing diffusion and obsolescence rates will have the effect of eroding the knowledge base of agents over time. All other parameters being equal we would therefore expect that agents subject to high obsolescence would have fewer surviving knowledge **links** than those experiencing low obsolescence. The same applies to agents experiencing high compared to low diffusion. The graph above shows that this is the case, showing that the simulation was performing as expected for our settings.

3. Average Revenues per Agent per Period at the end of 80 periods



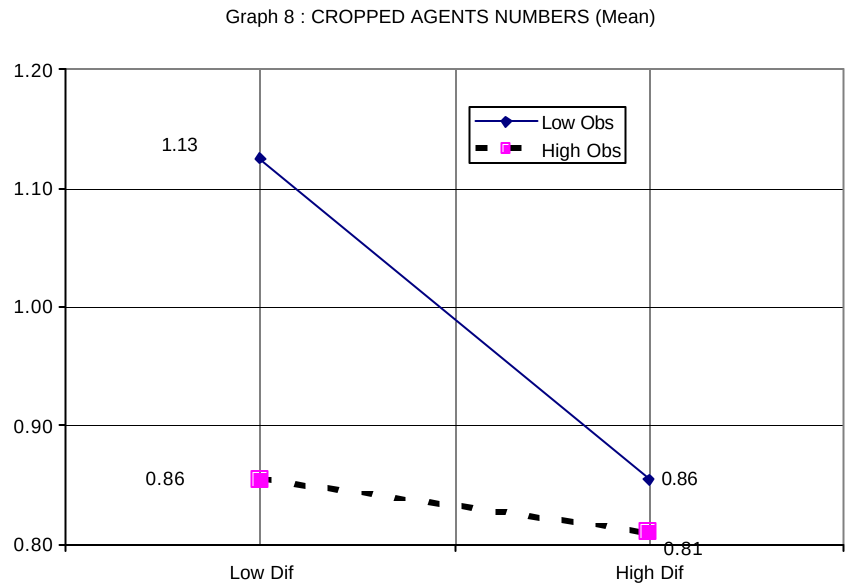
As obsolescence and diffusion rates increase the ability of the agent to sustain rents from their knowledge should erode, which means that at the end of the simulation the average revenues the agent can generate per period will be less for agents facing high diffusion regimes. The same applies to high obsolescence regimes. The graph above demonstrates that we get the expected results.

4. Average number of agents at the end of 80 periods



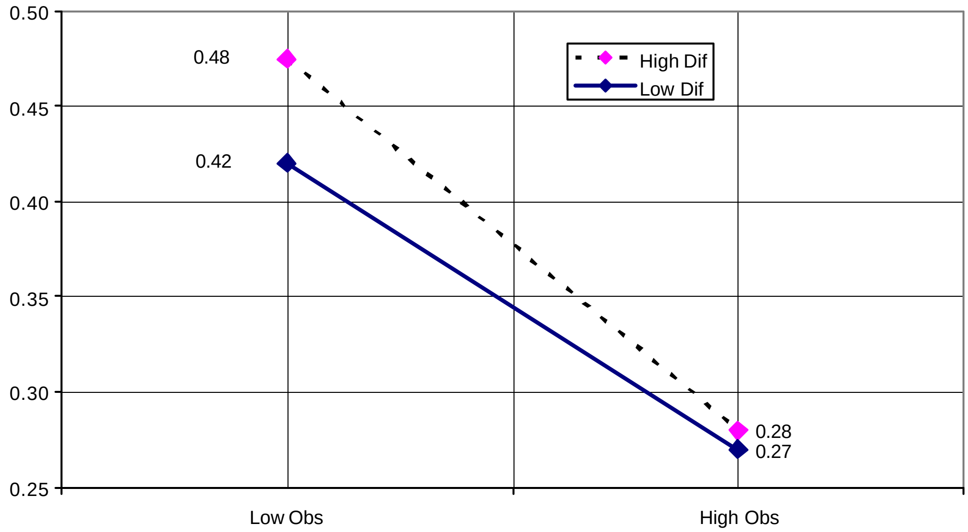
Because agents in high obsolescence environments capture less revenues and have less nodes and links to generate those revenues, the number of survivors will be lower compared to low obsolescence conditions. The same argument applies to high versus low diffusion. Our graph shows that we get the expected results.

5. Average Number of Agents being Cropped per period at the end of 80 periods



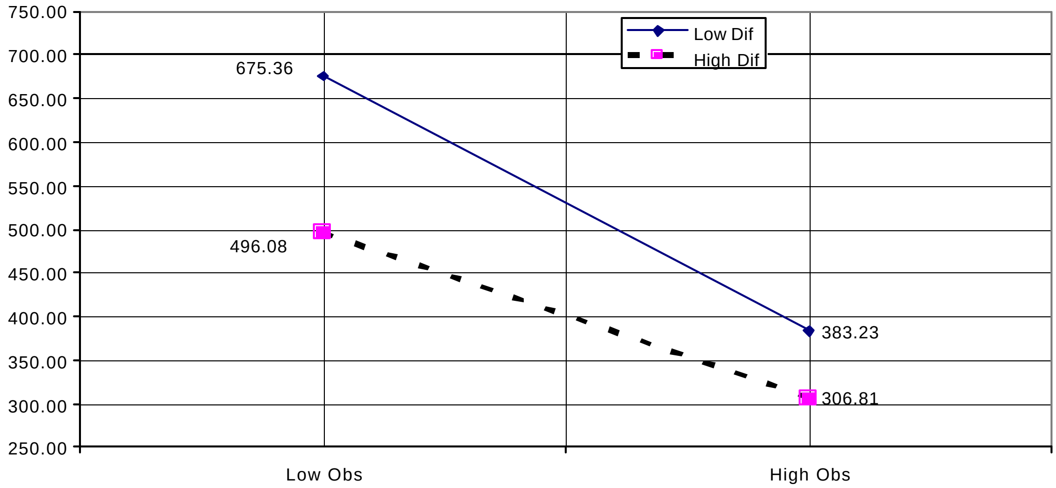
The graph unexpectedly shows that the number of agents being cropped per period is **higher** both for the low diffusion and for the low obsolescence cases**.** Although the actual number of cropped agents decreases with both increasing diffusion decay and obsolescence decay, we need to recognize that there are more agents that can get cropped in both the high diffusion and the high obsolescence cases. So, if we take agents cropped as a **percentage** of total surviving agents, the number actually goes up from 5.2% to 5.6% of total agents as diffusion increases; and from 6.4% to 7.6% of total agents as obsolescence increases. This result is in line with what one would expect.

6. <u>Average number of agents exiting per period at the end of 80 periods</u>



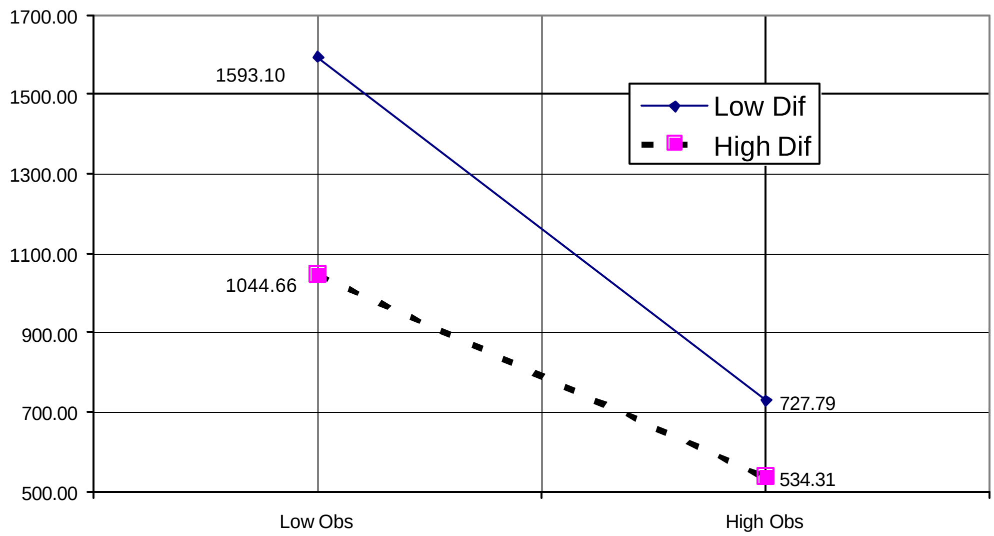
Unexpectedly again, the absolute agent numbers exiting are greater for both high obsolescence and high diffusion decay. But once again the story lies in the **percentage** Agents exiting. A larger percentage of agents (2.6 vs 2.2%) exit for high obsolescence conditions compared to low, with a similar difference for high diffusion compared to low.

7. Financial funds cumulated in the simulation over the last 40 periods of the 80 period simulation



Under conditions of both high obsolescence and high diffusion there are fewer agents each generating fewer revenues, so that over time the total financial funds agents can accumulate for meetings will be less under high diffusion conditions and also under high obsolescence conditions compared to their respective low conditions. The graph above demonstrate that we are getting the expected results

8. Experience funds cumulated in the simulation over the last 40 periods of the 80 period simulation



Under conditions of both high obsolescence and high diffusion there are fewer agents, each generating fewer revenues, so that over time the total experience funds agents can accumulate for investing in knowledge development will be less under high diffusion conditions and also under high obsolescence conditions compared to their respective low conditions. The graph above demonstrates that we are getting the expected results

Conclusion

The purpose of the paper was to describe in detail the verification process that we followed in developing the Sim-I-Space simulation model and partially validate the model by demonstrating that it produces results in line with the original model design.

We have documented the verification process that was followed and shown in the graphs and tables that as we adjust the levels of diffusion decay and obsolescence decay, we get expected patterns of outcomes for agent population variables, agent population dynamics, and cumulative financial conditions of these populations. This increases our confidence that the model is operating in accordance with its initial design.

APPENDIX 1: Model Validation by Tables

Table A1.1: Average financial conditions at the final period of the simulation (80th period - for 200 runs)

	Average rent per agent	Financial funds per surviving agent	Experience funds per surviving agent
MEAN			
OBSOLESCENCE	LOW HIGH	LOW HIGH	LOW HIGH
Low diffusion case	16.4>13.9*	30.8>25.6*	72.6>54.3*
High diffusion case	15.8>12.3*	30.4>28.6*	64.0>49.9*
DIFFUSION	LOW HIGH	LOW HIGH	LOW HIGH
Low obsolescence case	16.4>15.8	30.8>30.4*	72.6>64.0*
High obsolescence case	13.9>12.3	25.6<28.6*	54.3>49.9*

* Significantly different at $p<0.05$ based on the t-test comparing means.

As obsolescence decay increases rents per agent, financial and experience funds per agent all decrease. As diffusion decay increases, rents per agent, financial and experience funds all decrease. This is what we would expect if the simulation is working in a way that reflects its design. All differences are in ther right direction and most differences are significant

Table A1.2: Average agent conditions at the final period of the simulation (80th period for 200 runs)

	Number of agents		Number being cropped		Percentage being cropped		Number of agents exiting		Percentage exiting	
MEAN										
OBSOLESCENCE	LOW	HIGH	LOW	HIGH	LOW	HIGH	LOW	HIGH	LOW	HIGH
Low diffusion case	21.9>13.4*		1.13>0.86*		5.48<6.72		0.48>0.28*		2.35>2.36	
High diffusion case	16.3>10.7*		0.92>0.81		6.25<8.72*		0.42>0.27*		2.90>2.49	
DIFFUSION	LOW	HIGH	LOW	HIGH	LOW	HIGH	LOW	HIGH	LOW	HIGH
Low obsolescence case	21.9>16.3*		1.13>0.92*		5.48<6.25		0.48>0.42*		2.35<2.90	
High obsolescence case	13.4>10.7*		0.86>0.81		6.72<8.72*		0.28>0.27		2.36<2.49	

* Significantly different at p<0.05 based on the statistical mean analysis by t-test.

As obsolescence decay increases, number of agents created, numbers exiting and percentage exiting decrease, percent number of agents cropped increases but agents cropped decrease. As diffusion decay increases, agents created all decrease. This is what we would expect if the simulation is working in a way that reflects its design. All differences are in the right direction and most differences are significant

Table A13: Total financial conditions at end of simulation (80^{th} period for 200 runs):

	Financial funds of survivors	Total financial funds created	Experience funds of survivors	Total experience funds created
MEAN				
OBSOLESCENCE	LOW HIGH	LOW HIGH	LOW HIGH	LOW HIGH
Low diffusion case	675>383*	713>407	1593>728*	1653>764
High diffusion case	496>307*	528>328	1045>534*	1094> 67
DIFFUSION	LOW HIGH	LOW HIGH	LOW HIGH	LOW HIGH
Low obsolescence case	675>496*	713>528	1593>1045*	1653>1094
High obsolescence case	383>307*	407>328	728>534*	764> 567

* Significantly different at $p<0.05$ based on the statistical mean analysis by t-test.

As obsolescence decay increases financial and experience funds increase but financial and experience funds per agent decrease. As diffusion decay increases financial and experience funds increase but financial and experience funds per agent decrease. This is what we would expect if the simulation is working in a way that reflects its design. All differences are in the right direction and most differences are significant

Appendix 2: Variable Settings for Validation Runs

Global switches	
ASSET_MANAGEMENT	0
MOD_DISCOVERY	0
MOD_RESEARCH	0
AGENT_INTERACTION	0
MOD_OBSOLESCENCE	0
MOD_DIFFUSION	0
Main variables	
INIT_AGENT_NUM	20
INIT_NODE_NUM	15
INIT_LINK_NUM	10
MODEL_PERIODS	200
MAX_AGENT_NUM	200
MAX_NODE_NUM	20000
MAX_LINK_NUM	20000
Asset variables	
MAX_COMPLEXITY	7
COMPLEXITY_COST	0.001
NEW_LINK_PROB	0.3
MOVE_POSSIBILITIES	2
PASSIVE_CARRY_MULT	0.7
I-Space world variables	
BASE_REV_MULT	70
ASSET_SHARE_PROB	0.2
OBSOLESCENCE_DECAY	0.025
DIFFUSION_DECAY	0.025
DIFFBLOCK_COST_MULT	0.05
AGENT_ENTRY_THRESHOLD	0.01
AGENT_ENTRY_RATE	7
AGENT_EXIT_THRESHOLD	70

AGENT_EXIT_RATIO	-
MAX_AGENT_ENTRIES	3
AGENT_ENTRY_NODES	3
AGENT_ENTRY_LINKS	2
I-Space matrix variables	
ABSTRACT_DIM_SIZE	5
CODIFY_DIM_SIZE	5
DIFFUSE_DIM_SIZE	4
DIFFUSE_FACTOR	2
Linkage probability variables	
ABS_NONZERO_VAL	0.25
COD_INCREASE_VAL	0.25
ABS_NONZERO_PROB	0.06
NEW_ASSET_THRESHOLD	1.4
Agent variables	
INIT_FINANCIAL_FUNDS	20
INIT_EXPERIENCE_FUNDS	20
FINANCIAL_ALLOCATION	0.7
ACTIVE_SET	20
PASSIVE_SET	15
ISPACE_MOVE_PROB	0.03
ISPACE_MOVE_COST	0.03
JOINT_VENTURE_RETURN	0.1
SUBSIDIARY_RETURN	0.1
PAR_FUND_THRESHOLD	30
PAR_ASSET_THRESHOLD	2.8
Agent meeting	
MEETING_ARRANGE_COST	0.005
MEETING_FIXED_COST	0.005
PRESENT_COST_MULT	0.005
EXAMINE_COST_MULT	0.005
TRADE_VALUE_MULT	2.12

LICENSE_VALUE_MULT	2.12
Meetingspace variables	
PROB_RANDOM_MEETING	0.2
MAX_RANDOM_MEETINGS	3
TRADING_SET_RATIO	2
DM variables	
MIN_PREFERENCE_LEVEL	1
MAX_PREFERENCE_LEVEL	3
Research DM variables	
PROB_MOVE	0.2
Meeting DM variables	
PROB_POSITIVE PROB_NEGATIVE	0.05 0.2
HIGH_COOP_MULT	0.3
JOINT_VENTURE_INVEST	0.1
SUBSIDIARY_INVEST	0.1

Appendix 3: Version history of the Sim-I-Space

This tracks conceptual changes as well as any other substantive changes. Aesthetic changes will take place from version to version, but should not affect the substance of the document, and are excluded.

Changes from version 2.00:

1) Both financial and experience funds can be 2 types. One is survived only in the market, and the second is a cumulative one that includes exited and cropped agents.
2) The number of active set and passive set can be different. In the beginning, they should be the same numbers.

Changes from version 1.04:

1) Agents aggregate all encounters (arranged and random) every period and try to participate in the largest possible multi-Agent meeting it is able to attend.
2) Multi-Agent meetings only have Merger and Joint Venture as options. If the participating Agents choose not to Merger or create Joint Venture, the meeting breaks off into bilateral meetings (with presentation/inspection costs already prepaid).

Changes from version 1.03:

1) Added Agent interaction.
2) Only Nodes and Links of equal Complexity have any Linkage Probability. Thus, combination of Assets to create new Assets only takes place between Nodes and Links of equal Complexity.
3) Added Financial Budget and Experience Budgets, to make I-Space accounting more reflective of real-world accounting.
4) Clarify distinction between Nominal Diffusion – actual number of Agents owning the Asset, and Real Diffusion – the degree of Diffusion that makes a difference to the Base Revenue Potential of the Asset.

Changes from version 1.02:

1) Renamed the versions – all versions prior to Agent interaction are now versions 1.0x. In the earlier version, the Revenue Multiplier increased (at a decreasing rate) with Complexity. The negative return to Complexity is now formalized as the Carry Cost of the Asset, i.e. the net result is unchanged – the value of an Asset increases with Complexity, but the cost of maintaining it increases as well, thus beyond some level of Complexity, an Asset actually imposes a net cost on the Agent.
2) All “movement” (abstraction, codification, impacting, absorption) of Assets in I-Space is represented as the creation of new Assets in the relevant locations. Thus movement in I-Space can be seen as a chain of Assets of gradually Abstraction, Codification or Diffusion.

Changes from version 1.01:

1) Base Revenue Potential is now affected by a new variable Industry Multiplier that sets the Base Revenue Potential across industries for Assets of the same Abstraction, Codification and Diffusion value.
2) Added definition for Complexity, a rough measure for the complexity of the Node or Link.
3) The definition of Revenue Multiplier has been expanded to include the effect of Complexity, i.e. Revenue Multiplier increases with Complexity, to a point, beyond which it decreases (decreasing returns to Complexity).
4) Both Abstraction and Codification now affect the values in the Linkage Probability matrix: a) Abstraction increases number of non-zero values; and b) Codification increases value of non-zero values.
5) Added the definition of Impacting, which was left out.
6) Definitions of Obsolescence and Diffusion Decay now include the factors, which affect them. In this case, aside from Industry specific multipliers, Obsolescence is affected by number of new Nodes and Links generated in each period, and Diffusion is a function of the Abstraction and Codification of the Asset.

Changes from initial version (version 1.00):

1) Links are now discrete objects. In the earlier version, Links were a reference to the Linkage Probability between two nodes, such that Link #(15,25) would be the Linkage Probability between Node #15 and Node #25.
2) The axes of the Linkage Probability Matrix have been updated to Nodes vs. Links, to take into account the change above. As such, a Node will have some Linkage Probability with a Link, and Links are now to some extent "generic".
3) Updated the definition of Abstraction and Codification to distinguish between the effects on Nodes and Links – Codification of Links changes values in the Linkage Probability matrix, but Codification of Nodes do not.
4) Added the Example.