



Rainer Hegselmann (2012)

Thomas C. Schelling and the Computer: Some Notes on Schelling's Essay "On Letting a Computer Help with the Work"

Journal of Artificial Societies and Social Simulation 15 (4) 9
<<http://jasss.soc.surrey.ac.uk/15/4/9.html>>

Received: 18-Oct-2012 Published: 31-Oct-2012



Abstract

Today the Schelling model is a standard component in introductory courses to agent-based modelling and simulation. When Schelling presented his model in the years between 1969 and 1978, his own analysis was based on manual table top exercises. Even more, Schelling explicitly warned against using computers for the analysis of his model. That is puzzling. A resolution to that puzzle can be found in an essay that Schelling wrote as teaching material for his students. That essay is now made public by Schelling in JASSS, exactly 40 years after it was written. In his essay, Schelling gives a guided tour of a computer implementation of his model he himself implemented, despite his warnings. On this tour, though more in passing, Schelling gives hints to an extremely generalised version of his model. My article explains why we find the generalised version of Schelling's model on the tour through his computer program rather than in his published articles.

Keywords:

Schelling Model, Segregation, Configuration Game, History of Computational Social Science, Agent Based Modeling



Introduction

- 1.1 In a series of articles and books, published between 1969 and 1978, Thomas Crombie Schelling presented and analysed what is now known as the Schelling model (see Schelling 1969a; 1969b; 1971; 1972a; 1972b; 1974; 1978). The model addresses social segregation that "results from the interplay of individual choices that discriminate" (Schelling 1971, 143). The main focus of the model is residential segregation based on racial preferences. The model is extremely simple and abstract. It comes in two versions, as a 1-dimensional line and as a 2-dimensional checkerboard. The inhabitants of these one and two-dimensional worlds define their neighbourhoods in a spatially self-centred way: each individual's actual location is the centre of their neighbourhood. Individuals are content if their neighbourhood has their preferred colour mixture. If they are discontent they try to move to a more preferred location that matches their desired colour mixture. By manual table-top exercises, e.g. moving dimes and pennies on a checkerboard, the model shows that a massive residential segregation can result from only mildly segregationist preferences. It is sufficient that the individuals in the model wish to avoid minority status. The main and unexpected lesson is that massive segregation does not require strong segregationist preferences, i.e. preferences for living in a neighbourhood with high percentages of like coloured people. When the Royal Swedish Academy of Sciences announced on 10 October 2005 that Schelling had won the Nobel Prize for Economics (together with Robert Aumann), they cited Schelling's application of game theory to problems of arms race and international security as well as his models of segregation as the grounds for the award of the prize.
- 1.2 Over the years, Schelling's model has become a classical reference in many (partially overlapping) scientific contexts: explanation of residential segregation, unintended consequences, micro-macro relations, clustering, social phase transitions, invisible hand explanations, and emergence of spontaneous order. The model has also become a paradigmatic case study in the philosophy of science for both the study of mechanisms and the status of models more generally. Finally, and on that we will focus in the following, Schelling's model is considered as an early and pioneering example for an agent-based computer simulation.
- 1.3 Schelling's model has become a standard component (and appetiser) in introductory courses to agent-based computational social science. The model has been implemented on innumerable occasions and is a standard exercise for students. A cursory internet search highlights just how significant the model is for agent-based computer simulation. Given the bunch of lessons that we can take away from the model, nothing seems to be more natural than to implement and study the model on a

computer.

- 1.4** Yet, when we read what Schelling actually said about his model in the 1970s, we are faced with a perplexing puzzle. Writing about 40 years ago, he warns against using computers to analyse his model:

I cannot too strongly urge you to get the nickels and pennies and do it yourself. I can show you an outcome or two. A computer can do it for you a hundred times, testing variations in neighborhoods demands, overall ratios, sizes of neighborhoods, and so forth. But there is nothing like tracing it through for yourself and seeing the process work itself out. It takes about five minutes—no more time than it takes me to describe the result you would get. In an hour you can do it several times and experiment with different rules of behavior, sizes and shapes of boards, and (if you turn some of the coins heads and some tails) subgroups of nickles and pennies that make different demands on the color compositions of their neighborhoods (Schelling 1974, 48).

- 1.5** Schelling repeated such warnings several times and he meant them to be taken seriously. They sound absurd today, but at the time—the 1970s—they were not. A resolution of the apparent puzzle can be found in his essay "On Letting a Computer Help with the Work", written in 1972 and now published here for the first time by **JASSS**. Schelling used the essay as mechanically duplicated teaching material in courses at Harvard. At the start he tells us that it is "a guided tour through a computer program" (Schelling 1972a, 1). It is a tour through Schelling's own computer implementation of the 2-dimensional version of his model. (As it seems, the 1-dimensional version was never implemented by Schelling.)
- 1.6** Schelling's essay presupposes that the reader already knows the model. In section 2 I will provide a short summary of the 2-dimensional version of the model. In section 3 the perplexing puzzle is resolved. In section 4 we will see that Schelling's guided tour through his program is much more than that: it contains a vision of a fairly general model that goes far beyond what is now known as the Schelling model. In section 5, I conclude with an explanation of why it is no accident that Schelling's more general vision can be found in his guided tour through his programming of his model rather than in his other publications on the model.



A short summary of Schelling's Model

- 2.1** The most comprehensive and detailed description of Schelling's model can be found in Schelling (1971). It is an invited article for the first volume, issue 2, of the *Journal of Mathematical Sociology*. Except for a preface and a summary, the article is—as announced by Schelling in an annotation—basically identical with a memorandum that he had written for the RAND corporation (Schelling 1969a).
- 2.2** The decisive features of the 2-dimensional version are (Schelling 1971, 154ff.):
1. The playground is a finite checkerboard. The actual size is always 13×16 . Other checkerboard shapes and sizes are possible.
 2. There are two groups, graphically displayed as zeros and crosses, physically realised as coins, chips, counters, aspirins etc. The primary interpretation is that the tokens are people, belonging to two ethnic groups, blacks and whites. Other interpretations are possible, e.g. as boys and girls. The groups may be of different size. However, equal size is the starting point for the analysis.
 3. Normally the whole population gets scattered throughout the board by a random process. Starting with a specific constellation is mentioned as a possibility. Each cell can be inhabited by one and only one individual. About 25-30% of the cells remain empty to give enough clearance for movement.
 4. All individuals define their neighbourhood in terms of neighbouring cells that surround their actual position. The standard neighbourhood are the eight other cells in the 3×3 area around the cell in the centre of that area. But larger neighbourhoods, for instance the twenty-four cells of a 5×5 area, are mentioned as well. All individuals have a neighbourhood preference that states—in absolute or relative terms—the colour composition that they want to have in their neighbourhood. The preferences are defined in a variety of ways: The standard case is a minimum demand for like coloured neighbours, e.g. the requirement of not being the minority. Empty cells may be counted as others—or not (Schelling 1971, 165f). The standard case is that neighbourhood preferences are stated in terms of lower bounds. However, having both, a lower and an upper bound for like coloured neighbours, and a kind of scaled preferences over the possible colour mixes is considered. The neighbourhood preferences are the same within a group, but may differ between the groups.
 5. All individuals evaluate their neighbourhoods. Individuals whose neighbourhood does not accord with their preference are discontent. Otherwise they are content.
 6. The individuals can move. Different rules for the order of movement are mentioned, such as working from the upper left corner downward to the right, working from the centre outward or letting one group move first, or some random procedure.
 7. Content individuals always stay where they are, but a discontent individual moves to "the nearest spot that surrounds him with a neighbourhood that meets his demand" (Schelling 1971, 155). Distance between two cells is measured in terms of the smallest number of cells that one traverses (horizontally and vertically) to get from one cell to the other. Schelling gives no explicit specification for the case that there is no or more than one such nearest satisfying spot.

- 2.3** There is one finding of Schelling's model that is almost certainly always mentioned first. It is a striking macro effect, unintended

by the individuals in the model, and unexpected by the uninitiated observer of the model: Mildly segregationist preferences that demand not to be a minority in one's neighbourhood, are sufficient to generate a massively segregated checkerboard society. Slight segregation starts already with a demand for one-third of like coloured neighbours (Schelling 1971, 158).

2.4 The result occurs under the condition that the two groups are of equal size and with equal minimum demands of like neighbours. Schelling, however, found much more—and less known—interesting and unexpected results (Schelling 1971, 158-166). He experimented with equal numbers with unequal demands, unequal numbers with equal demands, congregationist preferences (they demand a certain absolute number of like neighbours, no matter whether the other neighbouring cells are opposite or empty), or integrationist preferences (they demand a certain ratio or come with both, a lower and an upper bound for the fraction of like neighbours).

- Under the condition of equal numbers the more demanding group ends up with an only slightly higher average proportion of like neighbours.
- If demands are equal, but one of the two colours is strongly outnumbered, segregation is much greater than in the equal number case).
- Being indifferent between empty cells and the other colour while requiring significantly less than a majority of like colour, causes almost the same degree of segregation as requiring a majority.

2.5 In (2006) Schelling complained a bit about the fact that the shortened version of his model that was published as chapter 4 (Sorting and Mixing: Race and Sex) of *Micromotives and Macrobehavior* (Schelling 1978) received much more attention than his original *JMS* article with the effect that some of the other results did not receive the attention he had hoped for.



An Easy Resolution of the Puzzle

3.1 On his guided tour, Schelling introduces the computer to the reader in an amusing way: the computer is an idiot:

I don't mean an ordinary idiot, I mean a "super-idiot". I mean an idiot with the characteristics that

1. he can perform a rudimentary task if he is told exactly what task to perform;
2. he can keep his place in an orderly list of instructions, doing the next task after he completes each one;
3. he will stop, and perhaps ring a bell, if the instruction is incomprehensible in his language or if the task he is told to do is unperformable;
4. he never makes a mistake or gets tired.

His reliability derives from two characteristics: he never makes a mistake in doing exactly what we tell him to do; and he never thinks for himself. (Schelling 1972a, 4).

3.2 Why, then, the warnings? What is wrong with such a speedy and reliable idiot? The answer is quite basic: The computer-idiot of the 1970s usually did not have an output device that could display and visualise an ongoing dynamics. The typical output device was not a screen, it was a tele-typewriter. Compared to the super-idiot, we would be better off with a real idiot:

If it were a real idiot pushing nickels and dimes around on a checkerboard, we could walk into the room and look at the checkerboard, copying or photographing it (Schelling 1972a, 33).

3.3 But the super-idiot of the 1970s could only send signals to a typewriter to print certain characters. That allowed to get out some statistics. It allowed for printouts of patterns of rows and columns in which group numbers indicate who is where on the checkerboard. But such printed pages are nothing that could be considered as a reasonable visualisation of the ongoing dynamics on the checkerboard. As Schelling writes in a personal communication:

In those days computers—at least, those I had access to—could compute outcomes but could not display the dynamics. I could run various hypotheses and examine the results, but not watch the process work. I was glad that I'd originally done it by hand, because I could see how things worked. That's why I may have advised doing it by hand on a table, at least to do it that way for a while to watch how things worked (Schelling pers. comm. 7.23.2012).

3.4 Additionally, it should not be overlooked that computing in those days required an enormous investment of time and energy. A print-out would require writing a program, producing punch-cards, and running the program in a certain time slot—often overnight—at a university computing centre, studying the error messages the next morning, debugging the program, producing new punch cards and so forth. All in all, do-it-yourself table-top exercises with moving pennies (or aspirins, as Schelling alternatively suggested) on large sheets of papers simply generated the "better pictures". A "real" computer was not necessary to get Schelling's impressive simulation results. Using a computer would endanger understanding. And all things considered, doing the simulation manually was faster.

3.5 The computational breakthrough occurred in the first half of the 1980s when the first mass-produced, affordable, and easy to use personal computers came on the market. Gone were the punch cards; in came a keyboard and mouse. Output was visible on a major screen and not only as a printout or a one-line-display. The screen could display graphics, doing away with the

need for a plotter. And, it should not to be overlooked, CPUs were now fast enough to calculate interesting social dynamics and their on-going graphical representation within a reasonable time.

3.6 All the obstacles and problems that had motivated Schelling to warn us against exploring his model with a computer were gone. From then on, whoever knew about the model, was interested to join the computational turn in social science, and was learning to program, almost unavoidably had the idea to implement Schelling's model as a computer simulation. Schelling's model quickly established itself as a classic in agent based modelling.

3.7 In an essay entitled "Some Fun, Thirty-Five Years Ago" Schelling (2006) wrote in the very last sentence:

Now that computers can display all the movement in 'real time' there is, I suppose, little advantage in doing this kind of thing manually.

a kind of 'official' correction of very reasonable warnings and recommendations given under other conditions some decades before.



On the Guided Tour: A Vision of a Generalised Schelling Model

4.1 Schelling's "guided tour through a computer program" (1972a, 1) is more than that. While he takes us through the components of the program, he hints at alternatives and extensions, that, taken together, amount to the vision of a generalised Schelling model. Although some generalising hints can be found in some of Schelling's published articles it is only on his guided tour that the generalised vision is stated almost explicitly—however, often in passing.

4.2 There are at least five generalisations in the essay. They are:

a. The group structure

The standard Schelling model has two groups. In the guided tour we have 1, 2, 3, ..., m groups.

b. The social space

In the standard model in Schelling (1971) the social space is a 2-dimensional checkerboard of size 13×16 . Later in Schelling (1978) it is an 8×8 checkerboard. In Schelling's computer program the size becomes $u \times v$. And, most importantly, Schelling considers social spaces of higher dimensions: "We could add dimensions, e.g., by doing floors in a multi-storied building, defining 'neighborhoods' and 'distances' appropriately" (Schelling 1972a, 40).

c. The neighbourhood concept

The standard Schelling model defines the neighbourhood of an agent as the 3×3 area around the cell he occupies. In the guided tour a whole bunch of neighbourhood concepts is offered: 5×5 , 7×7 , circular neighbourhoods are considered, distance may matter and more distant agents may be less important (Schelling 1972a, 13).

d. The preference structure

The guided tour describes an actual program, in which all other groups are taken as unlike. Neighbourhood preferences are defined as a percentages of like neighbours in relation to the occupied (!) cells in one's neighbourhood. In the description of the standard model, Schelling (1971) mentions other and more complex preference structures. For instance, there may be upper and lower bounds for the required number (or ratio) of members of the other group. Together with the other group, empty cells may count as unlikes, too. In the guided tour Schelling mentions another idea: degrees of likenesses between the groups and corresponding preferences (Schelling 1972a, 10, 40).

e. The rules of movement

The description of the migration regime in Schelling (1971) leaves questions open: Agents that are not satisfied, move to a spot, that meets their preference. But what if there is no such best place or more than one? On his tour through the program Schelling offers a collection of solutions to that and other open questions:

What we do is to have every individual consider his own neighborhood, decide whether he's satisfied with the mix of like and unlike neighbors and, if he's not, to consider moving. To consider moving, he looks around for blank spaces that he likes better than the position he is in. If he finds one nearby he moves to it. Or, perhaps, if he finds one anywhere he moves to it. Which one does he move to? Maybe he moves to the nearest one that is decently satisfactory. Maybe he moves to the best space on the whole board, regardless of how close it is (Schelling 1972a, 11).

We need a rule to cover ties. We can be arbitrary; keep the old one, new one, keep the nearest one to the individual on whose behalf we are searching the blank squares, or even pick a random number and take the new square if the number is odd and keep the old one if the number is even (Schelling 1972a, 27).

Maybe we drop the notion of satisfied and simply say that everybody compares every available vacant spot with the spot he's at, and moves to the best spot available if it's better than where he's at. Maybe we impose moving costs, expressed in a currency that is translatable into the attractiveness of different spots; and the moving cost is subtracted from the attractiveness in order to pick the spot that is best, considering both neighborhood and travel distance. And so forth (Schelling 1972a, 12; cf. 27).

4.3 Taken together, (a) to (e) clearly point into the direction of a generalized structure that might be called a configuration game (G, S, E, M), in which:

- G partitions the agents into m groups ($i = 1, 2, \dots, m$).
- S is a social space, given by a d -dimensional grid structure, or, and even more general, by any kind of connected graph (V, E) in which V is the set of nodes and E the set of connections between them. The agents inhabit cells or nodes and can move on the structure.
- E is an evaluation regime. It specifies how, based on this or that type of neighborhood preferences, the agents evaluate their actual and feasible alternative locations.
- M is a migration regime. It specifies, how agents get migration options and how they can use them. Given the evaluation of actual and alternative locations, principles formulate how to decide upon them—including how to break ties, or how, eventually, to make moves of despair if the status quo and the alternative locations are hopelessly bad in equal measure.

4.4 Such a generalized configuration game (G, S, E, M) is not explicitly stated in the guided tour. But (G, S, E, M) is a compact formulation of the abstract view or vision that seems to guide the guide on the guided tour through his program. [In Hegselmann and Kurz (2012) we work out in detail the idea of a generalized configuration game, and present some possible applications. Additionally, the (pre-)history of that idea is studied in a broader perspective.]



How Programming Changes the Programmer's Mindset

5.1 The computerisation of Schelling's model started very early. In the preface to Schelling's (now downloadable) RAND memorandum (Schelling 1969a), later published as Schelling (1971) without the preface, Schelling announces:

John Casti has been preparing a versatile computer program that will shortly give me freedom to explore a wider variety of hypotheses and that will generate more adequate examples for analysis. That will be another publication at a later date (preface, iv).

5.2 But, as both agree (Casti 1999; Schelling 2006), the cooperation did not work very well. Schelling's computational turning point was a session with his then Harvard student James Vaupel (Schelling 2006). In a personal communication, Schelling writes:

My original work was all table-top manual. I then asked RAND whether somebody might do a computer program, and John Casti got the job. It turned out not altogether satisfactorily; John and I hadn't worked together and there were misunderstandings, mostly mine. (For example, people counted as their own neighbors, people at the edges miscounted.) I finally got a student of mine, James Vaupel, to teach me how to program the analysis in Basic. We spent a Sunday afternoon with a blackboard, and he taught me how to do it. He left for Europe the next day and I was on my own. But I had learned, and from then on I could run the thing on a computer (Schelling, pers. comm. 7.23.2012).

5.3 James Vaupel insists, however, that it was a Saturday morning, not a Sunday afternoon. He continues that Schelling was "a very quick learner" (J. Vaupel, pers. comm. 27.7.2012). To start programming must have been a new experience for Schelling. In the preface to the guided tour he writes:

I have to add that one of the exquisite learning experiences of my life occurred one Sunday afternoon when, for three hours in a room with a blackboard, Jim Vaupel completely disassembled my 'model' into its smallest components and reassembled it before my eyes as a set of instructions that a computer could follow. (He sailed for Europe next day and I was on my own.) (Schelling 1972a, Preface 1f).

Was it just programming that Schelling started to learn on that day—be it a Saturday morning or Sunday afternoon? I believe, what he learnt was much more than programming. To my mind, it is no accident that we find a generalised version of Schelling's model, designed in passing, in his guided tour through his programming and not in his articles with all the illuminating results of manual table top exercises.

5.4 The point is that programming changes one's view on the subject the program is about. There are two reasons for that. One is very obvious and well known among programmers; the other is less obvious and deeper:

- First, programming requires a decomposition of a whole that is given before by a more or less intuitive description. Simple components have to be identified and specified. Their interplay, the parameters and their possible values, all that has to be exactly defined in a precise language with a rigorous grammar. Whoever started to program, realises after some lines of coding the ambiguities and the holes in informal descriptions that, beforehand, he or she considered

perfectly clear and complete. Programming implies to resolve the ambiguities and to fill the holes. In short: Programming forces the programmer to sharpen his or her view on the subject.

- Second, programming has an inherent tendency, if not irresistible seduction, to generalisation: it often transforms the original subject, its features or components, into instances of something much more general. A first example: In an informal description we may have two groups, blacks and whites. In the program they are represented by two lists. But why to stop with just two. Why not three, four, five ... ? Whoever commands the natural numbers, can't avoid asking that question, and in that moment the idea of a generalised group structure with m groups of possibly different group size is born. Such a generalised view may require not even a single additional line of code, because the original case (blacks and whites only), was technically realised in such a way that, whether two or m groups, is simply a question of just one parameter value in the lines that were already written to cope with two groups. It is a frequent programming experience that code, written to implement a specific feature of the subject matter, unintentionally (!) realises the particular as an instance of a generalisation that goes far beyond the original feature. Another example: While programming rules of movement, starting from some intuitive descriptions, it becomes obvious that there are much more and totally different possible rules. For instance, rules that require consent of the neighbourhood that someone wants to enter. Almost never can we implement all alternatives. But from now on we know that whatever we implement is just an instance of something more general that might be called a migration regime. Again, programming has changed the view. In short: Programming is an eye opener; by programming—for the most part unintentionally—we get to a more general point of view.

The first effect is demonstrated and stressed by Schelling again and again in his guided tour. All the ideas about a generalised version of his model, that Schelling gives on the tour, are beautiful evidence for the second effect.

5.5 As it seems, it pays off to program a computer to let him help with the work.



Acknowledgements

Many thanks to Thomas Schelling for the permission to publish his essay. I also want to thank James Vaupel for the hint that Thomas Schelling once had written an engaging paper on "how computers can help with the work" (pers. comm. 7.23.2012). Thomas Schelling was so generous to send me his own last copy of the essay. Mathias Risse helped to find another copy in Harvard university. That allowed to reconstruct half a page that, due to a production failure, was missing in Thomas Schelling's copy. Thomas Schelling, James Vaupel, and John Casti helped me in personal communications with answers to some questions.



References

- CASTI, J. L. (1994). *Complexification—Explaining a Paradoxical World Through the Science of Surprise*. New York: HarperCollins.
- HEGSELMANN, R. and Kurz, S. (2012). Schelling's Model Reconsidered, the Forgotten Pioneer James Minoru Sakoda and the Idea of a Generalized Configuration Game. [draft available from the author] [doi:10.1080/0022250X.1971.9989794]
- SCHELLING, T. C. (1969a). Models of segregation [RAND memorandum rm-6014-rc. Technical report]. Santa Monica, California: RAND Corporation. <http://www.rand.org/about/history/nobel/schelling.html> [doi:10.1080/0022250X.1971.9989794]
- SCHELLING, T. C. (1969b). Models of segregation. *American Economic Review* 59: 488-493. [doi:10.1080/0022250X.1971.9989794]
- SCHELLING, T. C. (1971). Dynamic models of segregation. *Journal of Mathematical Sociology* 1: 143-186. [doi:10.1080/0022250X.1971.9989794]
- SCHELLING, T. C. (1972a). On Letting the Computer Help with the Work [Teaching & Research Materials No. 12, Public Policy Program, John F. Kennedy School of Government, Harvard University].
- SCHELLING, T. C. (1972b). A process of residential segregation: Neighbourhood tipping. In Pascal, A. H., editor, *Racial Discrimination in Economic Life* (pp. 157-185). Lexington, Massachusetts: Lexington Books.
- SCHELLING, T. C. (1974). On the Ecology of Micromotives. In Marris, R., editor, *The Corporate Society* (pp. 19-64). London: Macmillan.
- SCHELLING, T. C. (1978). *Micromotives and Macrobehavior*. New York: Norton.
- SCHELLING, T. C. (2006). Some fun, thirty-five years ago. In Tesfatsion, L. and K. L. Judd, editors, *Handbook of Computational Economics* (Vol. 2)—Agent-based Computational Economics, volume II, (pp.1639-1644). Amsterdam: Elsevier.