Virtual Spiders Guide Robotic Control Design

Thiemo Krink, University of Aarhus
Fritz Vollrath, Oxford University

TO EXPLORE THE SPATIAL WORLD OF DIGITIZED SPIDERWEBS, THE AUTHORS MODELED THE GARDEN CROSS SPIDER AS A VIRTUAL ROBOT. USING LOCAL BEHAVIOR PATTERNS UNDER THE SUPERVISION OF A RULE-BASED SYSTEM TO CONTROL ARTIFICIAL LEGS AND SENSORS, THEIR APPROACH REQUIRED ONLY A FEW PATTERNS TO ACCURATELY GENERATE A LIFELIKE SPIDERWEB.

TRADITIONALLY, RESEARCHERS have studied natural science phenomena by describing observations and by experimentally isolating and analyzing pertinent parameters and factors. As computational power has mushroomed in recent years, analysis of biological systems has increasingly involved formal models coupled with computer simulations. Lately, investigators have turned to individual-based models that mimic the complexity of biological processes by using repeated local interactions between units, described in few simple behavior rules.[1] By mimicking plant morphology[2] or the problem-solving capabilities of "social" robots,[3] computer scientists can use a comparable concept for simulation, applied chaos theory, artificial intelligence, and artificial life.

In our approach, called TheseusV, we used this approach to mimic the spatial orientation of spiders during web construction. As this article shows, behavioral principles of arthropods (spiders and insects) during orientation are interesting not only for research into the behavior of real animals, but also because their robustness and simplicity make them potentially quite useful for controlling autonomous agents such as insect robots. After all, most real bugs are notoriously good at coping with unpredictable environments, so lessons we learn from animal models can also enhance artificial-life models and offer new insight into spatial orientation to AI researchers.

Why spiderwebs?

Spiderweb construction is a rare example of spatial orientation in biology that allows justified simplifications and abstractions as well as a straightforward model validation:

- first, the orientation and building procedure can be simplified as a 2D graphical task of short-range orientation,[4] and
- second, the entire behavior is reflected in the web, which can be graphically analyzed and compared with other webs.

In the first case, the spider's orb web is largely a 2D structure—the spider acquires the necessary spatial information through touch and internal body sensors. These sensors provide cues about contact with the web's threads as well as the position and movement of the spider's leg joints and current body posture.[5] Also, spiders measure and use geometric information, such as distances and angles between threads.[6] (Vision doesn't apply here because these spiders are almost blind and can construct a perfect web blinded or in total darkness.[7]) Spiders can also use gravity as an additional information source.[8]

Besides these important preconditions for proper computer simulations, the web-construction task is particularly interesting for studying spatial orientation because the animal must cope with a constantly changing environment.[9] During construction, the spi-

1094-7167/99/$10.00 © 1999 IEEE

der itself constantly inserts, tightens, and cuts threads, which it then uses for subsequent orientation and construction steps. Rain, wind, and trapped insects or wind drift (such as leaves) also break threads unpredictably.

Yet, despite this unpredictability, the spider's behavior almost always produces the typical web—a highly optimized structure that serves as a frozen trace of behavior patterns for our observations. Web construction is a rare example of an orientation process that takes place in a limited, readily observable, and freely manipulated arena. Thus, we can reasonably and easily validate our computer simulations of the spider's orientation behavior against real webs by comparing their geometrical designs.

Given the web-building process's robustness and local characteristic (use of local spatial information and local impacts of interaction, such as insertion of threads), we would expect to find emergent properties in the web's global pattern. Indeed, we identified one such property in the coiling pattern of the auxiliary and capture spiral of the garden cross spider. Our virtual robot's successful imitation of this pattern strongly supports the validity of our model,[10] which we previously used to tackle specific biological questions.[11,12]

Orb-web building of *Araneus diadematus*

We chose the garden cross spider *Araneus diadematus* because it builds a highly structured 2D orb web[13] and is easy to handle in the laboratory. To build its web, a garden cross spider first explores potential sites, before building a Y-shaped structure by lowering itself down on a thread line from the middle of a thread bridging two supports (see Figure 1). The junction of the Y becomes the web's hub; the arms and stem are the first radials. In stage 2, the spider finishes the hub by moving around the center, attaching more radials as it goes and setting up a surrounding framework of threads. Next, it constructs the auxiliary spiral, which consists of four to eight spiral turns from the hub toward the web's outside. In stage 4, the spider builds a sticky capture spiral from the outside to the center, which can consist of up to 50 turns, usually with several reverses (especially near the periphery). Finally, it fine-tunes tensions by pulling and interconnecting radials at the hub.

Modeling a virtual spider robot. Our study's primary goal was to develop a deeper understanding of the type and amount of information, memory, and intelligence needed for web construction by a relatively simple organism or machine (such as a spider or robot).

Our approach differs from earlier computer studies on spider behavior,[14] mainly because we tried to generate a web using a simulated spider robot that must explore, construct, and modify a spatial world with its biologically justified perceptive, cognitive, and motor abilities. Consequently, we did not give the virtual spider robot a priori spatial information—that is, our spider must explore, experience, and represent its spatial world itself. Thus, this approach resembles the situated behavior ideas proposed by Rodney Brooks,[15] who invoked autonomous agents that build their own world representation by experience and adaptation.

Also, this approach requires that the model's design be minimal (parsimonious): the virtual robot has as little cognitive abilities as necessary for it to perform. This minimal modeling is particularly useful when determining the indispensable memory requirements for web construction.

The "Related work" sidebar outlines further ways that our model relates to other AI approaches.

Design features. In designing a virtual spider as a simplified organism,[12] our methodology models each organ as an object accessible by a well-structured, constrained exchange of information aimed to not exceed a garden cross spider's abilities. Our model's simplifications and abstractions rely on previous investigations in web-building behavior and the physiological abilities of *A. diadematus*.[8,9,16,17]

Related work

Soar[1] is comparable to using an integrated AI system, which is designed to model human intelligence (psychological modeling) rather than animal orientation. Soar accounted for numerous psychological phenomena and relies on specific, cognitively motivated hypotheses about the structure of human problem solving. Its integrated conflict resolution by spanning and exploring an auxiliary search space added flexibility and generality, which our project did not require. Probabilistic reasoning, which has been successfully applied in medical-biological contexts (such as Mycin[2]), did not suit our project objectives either.

Indeed, our goal was to capture and represent a qualitative and causal interpretation of the web-building process made by a human observer. Our approach shows that a bottom-up representation of the behavior patterns and their interactions with the animal's motor-sensory system can do this. Our idea is much closer related to methods used in qualitative simulation,[3] which tries to mimic human ability to predict system behavior based on causal relationships, commonsense knowledge, and categories.

Generally, robotics approaches—particularly autonomous robots—deal with tasks comparable to those our virtual spider robot faced, although they are usually based on subsymbolic AI.[4] However, the scientific goal, and thus the modeling process, is very different. Robotics aims to build artificial agents and improve their performance to solve a specific task. By contrast, we wanted to imitate a natural organism to pinpoint missing and incorrect traits in scientific hypotheses, using a spider's decision-making process as its paradigm. We assumed that the complexity of spiderweb-building behavior could be generated as an emergent property of repeated interactions of simple units and local behavior patterns. Consequently, we have used a simple inference engine for the reasoning process (feedforward, procedural knowledge representation, and monotone reasoning), while the accurate representation of the spider's functional organs and their interactions was of major importance. For this, we combined object-oriented programming, symbolic AI, and qualitative simulation.

References

1. J.E. Laird et al., "Soar: An Architecture for General Intelligence," *Artificial Intelligence*, Vol. 33, No. 1, 1989.
2. B.G. Buchanan and E.H. Shortliffe, *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heurisitic Programming Project*, Addison-Wesley, Reading, Mass., 1984.
3. B. Kuipers, "Commonsense Reasoning about Causality: Deriving Behavior from Structure," *Qualitative Reasoning about Physical Systems*, D.G. Bobrow, ed., MIT Press, Cambridge, Mass., 1984, pp. 169–203.
4. D. Floreano and N. Stefano, "Adaptive Behavior in Competing Co-Evolving Species," *Proc. Fourth European Conf. Artificial Life*, MIT Press, Cambridge, Mass., 1997, pp. 378–387.

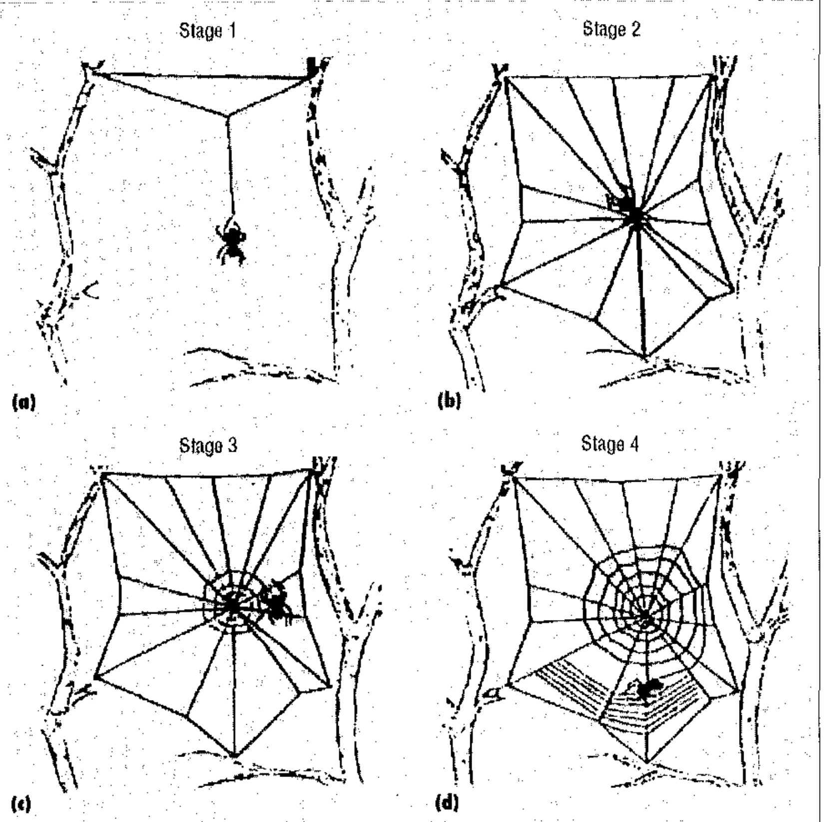

Figure 1. Web building of the garden cross spider *Araneus diadematus*: (a) in stage 1, the spider completes the hub and first radii; (b) next, in stage 2, it completes the frame, hub, and radii; (c) in stage 3, the insect undertakes auxiliary spiral construction from the center toward the peripheral web; (d) finally, the spider completes (sticky) capture spiral construction by circling back from the peripheral web to the center (stage 4).

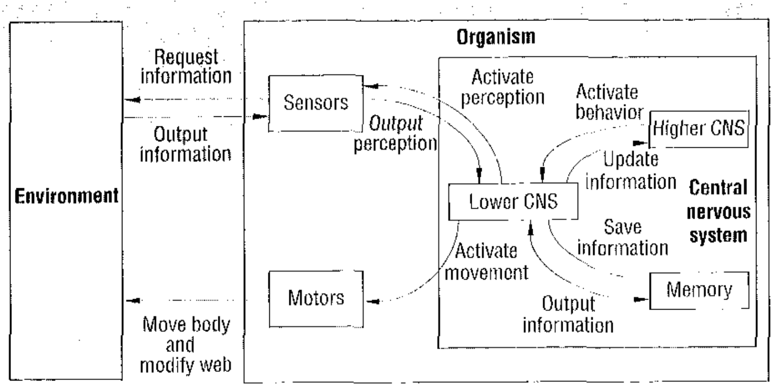

Figure 2. Components and communication of the simulated spider. The simulated spider has a central nervous system controlling sensors and motors. The CNS divides into a higher CNS (containing a rule-based system), a lower CNS (containing behavior patterns), and a memory. The organism interacts with the environment through its motor-sensory system, processing and storing perceived information for further actions.

The virtual organism consists of a central nervous system (CNS), motor abilities (motors) and a set of perceptive units (sensors) (see Figure 2). The heart of the CNS is a rule-based system (higher CNS) that contains our formalized hypotheses. This system's rules execute low-level behavior patterns (lower CNS), which interact with the virtual spider's environment by commanding the sensors and leg motors. We modeled only the two front legs of a real spider's eight legs because we assume those legs to be most important for tactile, motion-dependent, and body-internal perception. Our exclusive storage unit (memory) design made conservation and recall of information explicit.

We stepwise refined and validated the basic model in a four-step circular process:

1. We documented video sequences of web construction in detailed scripts and summarized them as mini-hypotheses.
2. Next, we formalized the extracted hypotheses as rules and low-level behavior patterns communicating with sensors, motors, and a memory.
3. Third, the virtual robot constructed a capture spiral and dismantled the auxiliary spiral of a real garden spider's half-done digitized web.
4. Finally, we compared the artificial capture spiral with the original one.

Simulation failures pointed to incomplete or inconsistent aspects of our hypotheses, thus forcing us to focus again on particular aspects of real web building and to modify the model.

Dealing with space and time. Knowledge representation of spatial orientation requires both time and space imaging. With the virtual spider, an accurate image of real-life motor actions requires a representation of highly dynamic interactions involving sensors, motors, and cognitive control. Either the rules or the behavior patterns thus must deal with time. The rule-based system offers two methods for time representation.

The first option is on the level of the fixed-action behavior pattern by summarizing action sequences into time-indivisible patterns with event-dependent interrupts. For example, consider the action of stretching one leg along a thread until either a crossing thread is perceived or the leg cannot stretch any further.

The second option is an explicit representation of simultaneous processes on the rule level. The rule system's syntax is hierarchically structured by the concepts of sequences, competitions, and performance (see Figure 3). Sequences are ordered lists of rules that are meant to be executed in one uninterrupted flow. They are grouped into so-called `RuleCompetitions` that can be activated alternatively and compete for firing. On the top level, `RulePerformance`, different groups of competing rules execute simultaneously by multitasking.

```
Memory read         := "ruleSystem  getFact:" SMALLTALK symbol
BPActivation        := "ruleSystem doAction:" SMALLTALK symbol1
                                 ["withLeg:" SMALLTALK symbol2]
Memory write        := "ruleSystem  putFact:" SMALLTALK symbol1
                                    "value:" SMALLTALK symbol2
PreCond             := Memory read
Action              := BPActivation | Memory write
Rule                := PreCond [" " PreCond]* " " Action [" " Action]*
RuleSequence        := Rule [" " Rule]*
RuleCompetition     := RuleSequence [" " RuleSequence]*
RulePerformance     := RuleCompetiton [" " RuleCompetition]*

Example:       RP( RC( RS(R2 R4 R7)
                       RS(R3)       )
                   RC( RS(R5 R6)    ) )

In Smalltalk: rp := #( ( (ThAuxRule ThCapRule)
                         (ThRevRule)           )
                       ( (ThCutRule)           ) )
```

Figure 3. The rule system's formal framework. A syntactical description of the rule system consists of the concepts of sequences, competition, and performance. On the sequence level, ordered lists of rules execute in chronological order without interruption. Sequences are grouped together as a set of activation descriptions that can be scheduled at the same time and compete for firing (`RuleCompetition`). On the performance level, different groups of competing rules execute simultaneously—that is, they are time-interrupt controlled.

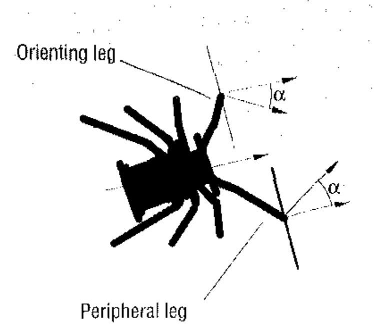

Figure 4. Motor direction commands. The virtual spider's front legs are orienting or peripheral according to their role for the orientation. Angle α indicates a change of a motor direction, specifying the relative direction change from the current body angle. The orienting leg interprets α mirrored—with a negative sign—in contrast to the peripheral leg. Thus, both legs can receive the same set of commands for turns toward and away from the body.

Implementation method. Our individual-based approach's design and implementation requires a computing design of individual components and clearly defined communication protocols for interactions.

Object-oriented languages can fulfill these requirements. Smalltalk, in particular, is a pure object language that provides a transparent view on single objects as well as the information flow between them. Powerful object browsers with classification aids for objects and methods allow quick changes of views on the model from greater distance and in close-ups.

Therefore, we used Smalltalk (VisualWorks 2.x on an Apple Power Macintosh) for our implementations; because we gave accurate design higher priority, we were willing to accept the drawback of poor execution-time efficiency compared to C++ or CommonLisp's CLOS, for example.

Specifying the orientation task. We decided to focus our simulation on the capture spiral construction, which is particularly interesting for several reasons:

- The finished capture spiral can be accurately described by geometric variables, thus letting us compare different spirals.
- These characteristics reflect all previous construction stages and are highly important for the web's efficiency in its function as a trap.
- Finally, this task is especially challenging because the spider dismantles threads of the auxiliary spiral that it previously used for orientation.

In our simulations, the virtual spider robot explores and cultivates its world on a graph representation of a digitized photo of a half-done web taken before the capture-spiral stage has begun. By giving the virtual spider preconditions comparable to those that a real spider would encounter, we aimed to simulate a realistic orientation task.

The robot's world. As a virtual spider robot

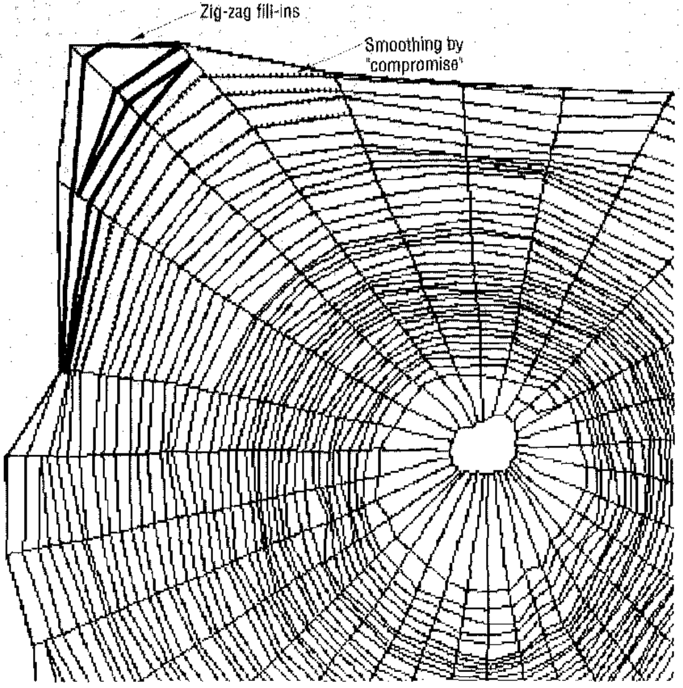

Figure 5. Web-reparation strategies. The garden cross spider tends to use reverses to fill out eccentric parts of the frame with sticky silk, especially near the web's outer periphery. As soon as the construction of the capture spiral becomes possible without severe deviation from a circular spiral, the spider begins to fix irregularities by a different method. Instead of repeated reverses (analogous to patching up a hole in a sock), the spider compromises, alternatively keeping the gap size constant between spiral turns and creating a circular spiral.

experiences its world, specific problems arise involving subjective interpretation of perceived information and body-organ coordination of motor actions. In particular, the modeled legs receive spatial information and perform actions relative to the current body orientation.

We implemented a direction control that specifies directions as distal, peripheral, (body-)forward, and (body-)backward (see Figure 4). It thus allows symmetric movement instructions independent of whether a leg is attached to the body's right or left side. In fact, symmetric commands are plausible for real spiders—it is unlikely that the order to contract a leg differs for right and left legs.

There is an additional reason for a complete abstraction of dealing with left and right legs arising from the spider's orientation along the auxiliary spiral. During the capture spiral's construction, the spider reverses its body direction several times, switching the body side that points toward the guiding auxiliary spiral. This switching would require that the spider switch the way it addresses the left and right leg after each reversal, because of their changing role for orientation. A more elegant solution involves communicating through a concept of peripheral searching and orienting along guideline legs mapped to left and right legs. The initial assignment is made when construction begins and swapped with each reversal. A real spider's initial peripheral-searching leg is always the leg that has been used for orientation while constructing the auxiliary spiral. Thus, the virtual spider receives this information as start-up information from the simulation setup.

Design properties of capture spirals. From the viewpoint of evolutionary theory, we can reasonably assume that the spider's goal of capture-spiral construction is the weaving of an optimal trap. Therefore, the spider must fill out the given space (bordered by the previously constructed frame) with an efficient prey filter of sticky threads.

We hypothesize that the spider must deal with two optimization tasks:

1. It must determine an optimal size of the gaps between spiral turns (called mesh or spiral distances). They must be small enough to catch maximum prey for the spider, but must also be large enough to reduce the amount of silk and time needed for the construction.
2. It must provide a regular gap size and make maximum use of the available space within the previously constructed frame.

By constructing a perfect circular spiral, the spider would reach the first goal, but for the latter goal, it must follow the frame's shape, which usually differs (and can differ considerably) from a perfect circular spiral.

The spider compromises by initially following the frame, consecutively and smoothly changing the spiral turns into a more circular shape the closer they approach the web's center. Most capture-spiral irregularities therefore arise in the peripheral web area. The spider tries to compensate for these irregularities by two behavior patterns: first, in case of larger deviations from a circular shape, by repeated reverses of its moving path, and second, by compromising between the generation of regular spiral distances and the attempt to design a circular-shaped spiral (see Figure 5).

We modeled our virtual spider robot on our observations of these compensatory behavior patterns and the spider's attempt to get as close to the frame as possible.

From observations to implementation

The spider's decision to reverse its moving path crucially determines the capture spiral's design. Based on a series of videotaped observations, we proposed a simple hypothesis, which we have implemented as one rule and two behavior patterns in Smalltalk syntax (see Figure 6). We quantified the expression "drastically reduced" with a specific behavior pattern (`ThBConsiderReverse`), which bases its (Boolean) decision on a (float) threshold

Hypothesis for Reversing (scientific description):
"Reverses occur if the current spacing of the inner leg's hold on the auxillary spiral to the outer leg's reach for the capture thread attachment point is drastically reduced compared to the previous spacing."

Reverse hypothesis (SMALLTALK syntax):

Rule:

```
ThRBeginReverse

activate

     self class registerPerformance.

     ((ruleSystem getFact: #actionStatus) = #muchSpaceLost) |
     ((ruleSystem getFact: #actionStatus) = #gapAhead)
     ifTrue:
        [self class registerFiring.
        ruleSystem doAction: #ThBReverse.
        ruleSystem putFact: #actionStatus value: #spiralTracking.
        ^true]
     ifFalse:
        [^false]
```

Behaviour patterns:

```
ThBConsiderReverse (activated after constr. of a new thread)

activate

     self class registerPerformance.

      (lowCog getFact: #lastCapPropDist) value >
      ((lowCog getFact: #capPropDist) value +
      (lowCog getFact: #lostSpaceThreshold) value)
     ifTrue: [lowCog putFact: #reverseStatus value: #muchSpaceLost].

ThBReverse

activate

          self class registerPerformance.

          lowCog changeLegFunctions.
          lowCog body turnAround.
```

Figure 6. This hypothesis implementation example shows how we transformed a scientific hypothesis about the spider's path reverses into formal rules and behavior patterns.

parameter `lostSpaceThreshold`. We captured the change of the distance between auxiliary and capture-spiral thread between two successive construction steps as `capPropDist` and `lastCapPropDist`, both measured and memorized. This rule performs the reverse if the previous patterns signal a "drastically reduced" distance (`muchSpaceLost`) by executing a motor action pattern (in this case, `ThBReverse`). The corresponding behavior pattern changes the roles of the leg functions: the innerLeg becomes the outerLeg and vice versa, and the (passive part of the) body turns around by a full 180 degrees.

The tactile task of discovery and cultivation. Figure 7 shows the currently used (complete) set of rules and associated behavior patterns.

The first group of rules for basic orientation and movement contains three rules for following an auxiliary spiral threads (`TrackAuxSpiral`) and radials toward either the web center (`TrackRadiusCentral`) or the peripheral web (`TrackRadiusPeripheral`). The next two rules `MoveIn` and `MoveOut` let the spider robot change the guideline by moving on auxiliary spiral turn inwards or outwards. Finally, the `BeginReverse` rule schedules a movement reversal.

A second special group of advanced information acquisition forms by the explicit storage of information about the most recently passed auxiliary spiral crossing a radius (`RegAuxCross`) as well as two rules for a modified adjustment of the gap size between spiral turns meshes (`BottomMeshVariability` and `TopMeshVariability`).

In addition to these rules for spatial orientation, a set of three rules controls web modification by insertion of new threads under the condition of a previously perceived outer sticky thread (`SetCapNode`) or the special case of a perceived frame thread (`SetRestCapNode`). The latter is only active during the construction behavior's early stages.

Finally, we provided one rule to detect the end of the construction and to stop the web-building behavior (`Finish`).

Model validation by hypotheses testing. Figure 8 illustrates two characteristics of (both real and artificial) webs that we extracted and analyzed.

On four orientations: north (up), south (down), and west and east (sideways), we measured distances between the points where successive spiral turns meet the same radius (spiral distance). We also counted the number of (body orientation) reverses. We grouped web characteristics located in the capture spiral's outer-half spiral and separated them from those of the inner half. If our web-design hypothesis is valid, the variances of spiral distances and the number of reverses should be significantly higher in the outer web.

To calculate real and simulated webs, we used a two-way analysis of variance (Anova) for the spiral distances and a Chi-square test for the reverses after checking that the underlying assumptions for normality and equality of variances held. Sample sizes were $n(real) = 8$ and $n(simulated) = 8$.

Spiral distance. Figure 9a shows the results of the analysis of spiral distances, for spiral spacings along four radii closest to the directions north, south, east, and west. A comparison of variances of spiral distance (all four orientations pooled) differed significantly between inner and outer web areas ($F = 4.8345$; $P = 0.0363$), whereas we found no significant difference between real and simulated webs ($F = 2.5153$; $P = 0.1240$). The interactions were not significant ($F = 0.5784$,

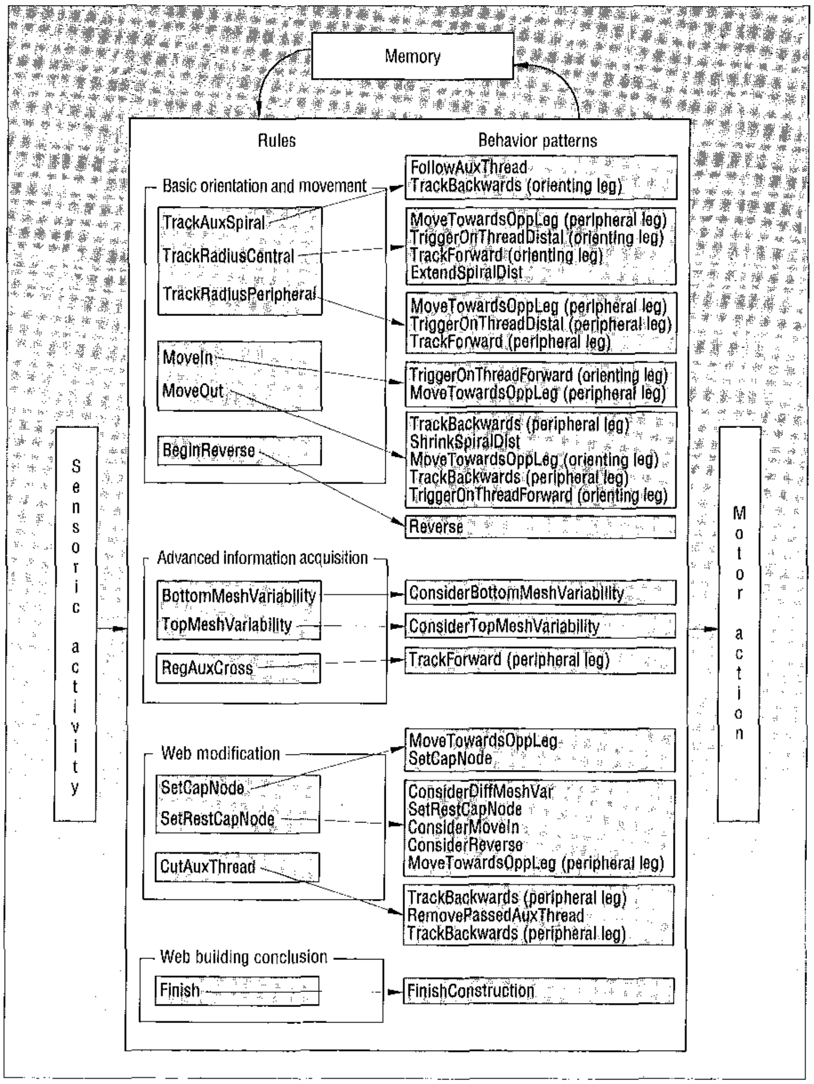

Figure 7. The simulated spider robot's rules and behavior patterns. Rules fire according to preconditions involving information coming from the memory and sensory activity of the touch, proprio, and gravity sensors. Each rule activates a set of behavior patterns that performs the wanted abstract instructions by coordination of motor actions.

$P = 0.4533$). (Note that F represents the value of the F distribution (named after R.A. Fisher) which, in this context, is used to test heterogeneity among sample means. P is the statistical probability that this heterogeneity has been misleadingly identified by chance. Important threshold values are: $p < 0.05$ statistically significant; $p < 0.01$ statistically highly significant; and $p < 0.001$ statistically very highly significant. For further reading on biologically relevant statistics see, for example, R. Sokal and F. Rohlf, *Biometry*, Freeman and Co., New York, 1995.)

Number of reverses. Figure 9b shows the analysis of body direction reverses. We found a highly significant difference was found between inner and outer web areas ($\chi^2 = 16.689854$; $P < 0.0001$). The comparison of real and simulated webs was, again, not significant ($\chi^2 = 0.543328$; $P = 0.4611$). The interactions were again not significant ($\chi^2 = 0.000059$, $P = 0.9939$).

Discussion

In applying individual-based and minimal (parsimony) modeling to the design of a virtual spider robot, we wanted to maximize the likelihood that our assumptions would reflect the set of strategies used by our real spider model. This is essential because model validation obviously can never prove that simulation results are caused by exactly the same principal as the characteristics observed on the system analyzed. Furthermore, we used our rule-based system for hypothesis falsification; we hoped to guide our research focus by pointing to incomplete or inconsistent knowledge.

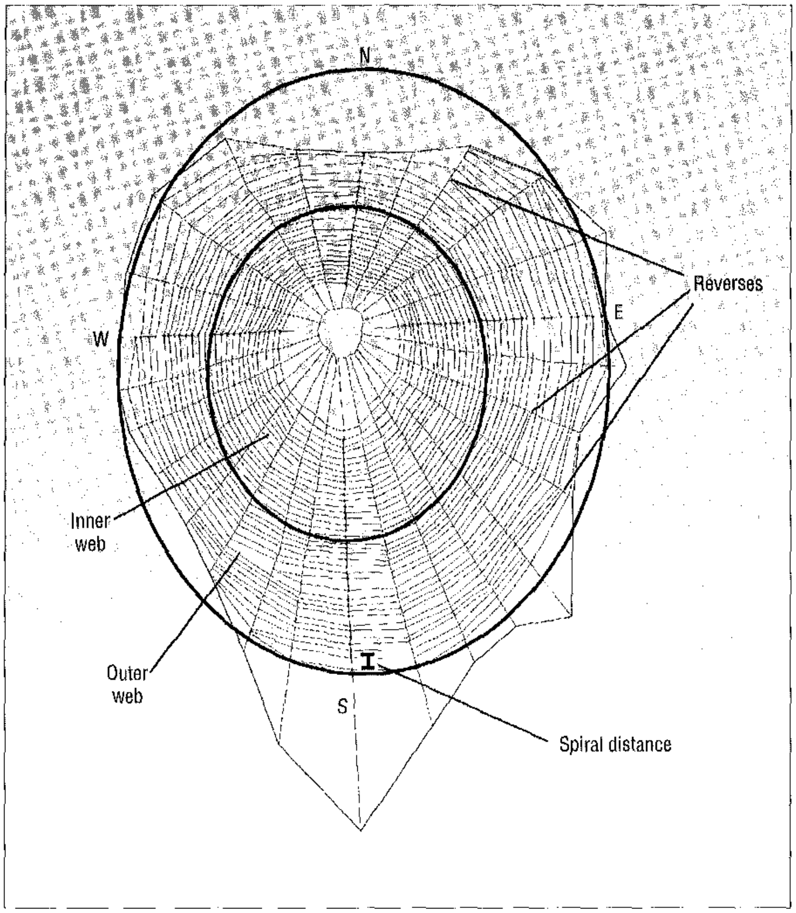

Figure 8. Web characteristics used for model validation. Spiral distance measurement along the four spokes N, S, W, and E, and number of reverses separated into inner and outer web. The inner web is the area of the capture spiral, which is within the range of half of its width.

We found that a virtual spider equipped with a small number of rules can construct a web similar to a real spider. We found evidence for the hypothesis that orb-web spiders might try to optimize the absolute size as well as the variation in size of spiral distances while using the available space as much as possible. The webs generated by the virtual spider robot shared characteristics with the real webs, thus supporting our mini-hypotheses of behavioral patterns of spatial orientation and construction.

Our example of spiderweb construction showed that, up to a certain degree of disturbance, even challenging tasks of spatial orientation could be managed by sequences of purely local-orientation patterns preceded by a few start-up conditions. Thus, in some cases, we can substitute global world representation in cognitive maps with an orientation behavior raised by the dynamics of repeated interactions between simple units. For a further investigation of this hypothesis, we are currently transferring this virtual approach to hardware robots. However, for the conceptual scientific questions we investigated here regarding animal decision making and spatial orientation, a simulation approach is more appropriate than a hardware implementation: in a robotics approach, we must deal with the robot's mechanical characteristics (highly specific) rather than the real animal's anatomy and physiology (with very different specifications).

OUR INVESTIGATION OF SPIDER web-building behavior has shown that virtual robots are powerful tools in artificial ethology and classical animal behavior studies. Given its open and transparent object-oriented design, this approach can easily be adapted to tackle a great diversity of behavioral questions. For the computer scientist, it might be equally interesting to employ this approach to unravel and identify novel, robust, and efficient solutions for use in hardware robotics. ■

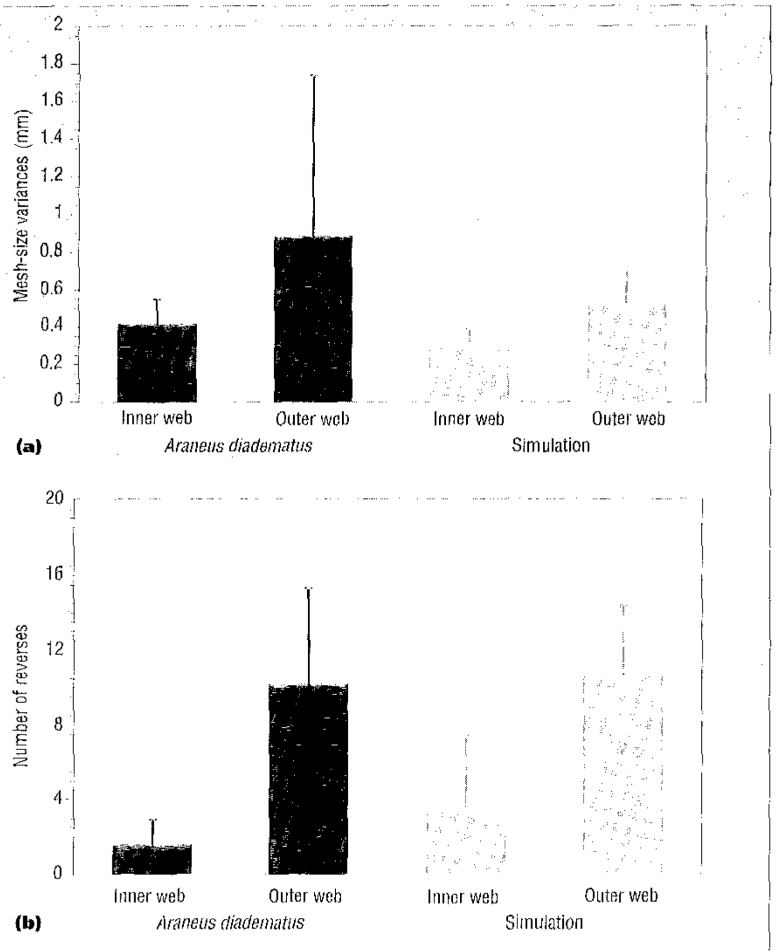

Figure 9. Comparison between real and artificial webs: (a) comparison of mesh-size variances between real and simulated webs grouped for inner and outer web areas; (b) the same comparison for reverses.

Acknowledgments

We thank Jacob Koella, Jutta Schneider, and an anonymous referee for comments, critique, and discussion.

References

1. O.P. Judson, "The Rise of the Individual-Based Model in Ecology," *TREE*, Vol. 9, No. 8, 1994, pp. 299–300.
2. D. Barabe, "Chaos in Plant Morphology," *Acta biotheoretica*, Vol. 39, No. 2, 1991, pp. 157–159.
3. K. Dautenhahn, "Getting to Know Each Other– Artificial Social Intelligence for Autonomous Robots," *Robotics and Autonomous Systems*, Vol. 16, Nos. 2–4, 1995, pp. 333–356.
4. T. Krink and P. Fuchs, *Modeling as a Tool for the Analysis of Spatial Orientation by the Example of the Web-Building Behavior of the Cross Spider*, MSc thesis, Hamburg Univ., Hamburg, Germany, 1994.
5. E.-A. Seyfarth, W. Eckweiler, and K. Hammer, "Proprioreceptors and Sensory Nerves in the Legs of Spider, *Cupiennius salei* (*Arachnida, Araneida*)," *Zoomorphology*, Vol. 105, No. 3, 1985, pp. 190–196.
6. F. Vollrath, "Altered Geometry of Webs in Spiders with Regenerated Legs," *Nature*, Vol. 328, No. 6127, 1987, pp. 247–248.
7. P.N. Witt, C.F. Reed, and D.B. Peakall, *A Spider's Web: Problems in Regulatory Biology*, Springer-Verlag, Heidelberg, 1968.
8. F. Vollrath, "Gravity as Orientation Guide During Web-Construction in the Orb Spider *Araneus diadematus* (*Araneae, Araneidae*)," *J. Comparative Physiology A*, Vol. 159, No. 2, 1986, pp. 275–280.
9. F. Vollrath, "Analysis and Interpretation of Orb Spider Exploration and Web-Building Behavior," *Advances in the Study of Behavior*, Vol. 21, No. 1, 1992, pp. 147–199.
10. T. Krink and F. Vollrath, "Emergent Properties in Orb Webs of Real and Virtual Spiders," *Proc. Royal Society*, Royal Soc., London, Vol. 265, No. 1410, 1998, pp. 2051–2055.
11. T. Krink and F. Vollrath, "Analyzing Spider-web-Building Behavior with Rule-Based Simulations and Genetic Algorithms," *J. Theoretical Biology*, Vol. 185, No. 3, 1997, pp. 321–331.
12. T. Krink and F. Vollrath, "A Virtual Robot to Model the Use of Regenerated Legs in a Web-building Spider," *Animal Behavior*, Vol. 57, No. 1, 1999, pp. 223–241.
13. F. Vollrath and W. Mohren, "Spiral Geometry of the Garden Spider's Orb Web," *Naturwissenschaften*, Vol. 72, No. 12, 1985, pp. 666–667.
14. W.G. Eberhard, "Computer Simulation of Orb-Web Construction," *Am. Zoology*, Vol. 9, 1969, pp. 229–238.
15. R.A. Brooks, "Intelligence without Representation," *Artificial Intelligence*, Vol. 47, Nos. 1–3, Jan. 1991, pp. 139–159.
16. R. Foelix, *Biology of Spiders*, Belknap Press of Harvard Univ. Press, Cambridge, Mass., 1982.
17. W.G. Eberhard, "Memory of Distances and Directions Moved as Cues during Temporary Spiral Construction in the Spider *Leucauge mariana* (*Araneae: Araneidae*)," *J. Insect Behavior*, Vol. 1, No. 1, 1988, pp. 51–66.

Thiemo Krink is an assistant professor and scientific coordinator of an interdisciplinary project at the University of Aarhus on the mutual influence of information technology and life sciences (EVALife). His research interests are principal mechanisms in behavioral ecology and evolution concerning the emergence of self-organization and robustness in complex biological systems, as well as transferring these results to algorithmic models and applications in adaptive and distributed computing. He received an MSc in computer science from the University of Hamburg and a PhD in biological science from the University of Aarhus, Denmark. Contact him at the Univ. of Aarhus, Ny Munkegade B540, DK 8000 Århus C, Denmark; krink@daimi.au.dk; http://www.daimi.au.dk/~krink.

Fritz Vollrath is a zoology professor in the Department of Zoology at Universitetsparken B135 and a visiting professor in the Department of Zoology at Oxford University. His research interests include animal behavior and orientation, in particular spiderweb architecture and evolution. He received an MSc and a PhD in biology from the Universities of Göttingen and Freiburg. Contact him at Oxford Univ., Dept. of Zoology, South Parks Rd., Oxford OX1 3PS, England; fritz.vollrath@zoo.ox.ac.uk; http://www.biology.au.dk/zoology/persons/fritzvolrathgb.html.