State Space Grids

Tom Hollenstein

State Space Grids

Depicting Dynamics Across Development

Springer

Tom Hollenstein
Queen's University
Department of Psychology
Kingston, ON K7L 3N6
Canada

ISBN 978-1-4614-5006-1 ISBN 978-1-4614-5007-8 (eBook)
DOI 10.1007/978-1-4614-5007-8
Springer New York Dordrecht Heidelberg London

Library of Congress Control Number: 2012948284

Printed on acid-free paper.

springer.com

Preface

It began in 1989. I had just graduated from the University of Massachusetts full of ideas and contemplating my future. Like many people, I had been captivated by James Gleick's book Chaos: The Making of a New Science. I immediately could foresee that the systems concepts emerging from physics and mathematics were applicable to psychology and human behavior. Here was a new way of thinking about causation in biological systems that fit more parsimoniously to my experiences as a person of the world and a student of psychology. That thoughts, emotions, and actions emerged or self organized simply from the interactions among lower-level components was beyond a theoretical model. It was almost metaphysical, like seeing the true nature of existence after only viewing Plato's shadows on the cave wall. My disappointment with the business of psychology I had felt as a student now gave way to hope and promise and excitement. I was turned on by this very large idea.

I went to the library to see if anyone had made the connection between psychology and chaos, fractals, or self-organization. Admittedly, my search was not exhaustive, but there was little research at the time and almost all of that was focused on physical action—finger wagging and such. Hence, I found no obvious champions that could possibly guide me through a graduate program in areas of psychology I found compelling, one of the possible futures I was considering at the time. Instead, I kept reading popular science books but left the hallowed halls of academia to travel and pursue other learning experiences. However, the seeds were sown and took more than a decade to germinate.

By September of 2000, I had been working at the Oregon Social Learning Center (OSLC) for over 4 years. I had advanced through the ranks to become the observational data analyst for the whole center. Sometimes I was extremely busy, and at other times the requests for analyses were not frequent or demanding. During these lulls, I was free to pursue other analytical techniques with the data. OSLC had a mountain of coding data from over 20 years of research on parent-child and peer interactions. I had actually first been hired as an observational coder applying the Interpersonal Process Code to the behaviors of parents and children. While watching hundreds of different families, the ideas of complexity, self-organization, and fractals resurfaced. Now, as an analyst, I was free to explore. As it turned out, several senior researchers at OSLC were familiar with and interested in applying systems approaches to better understand the family dynamics that led to behavior problems in childhood and adolescence. So, for about a year, I had periodic meetings with Jerry Patterson, Mike Stoolmiller, and Tom Dishion where I would present my latest attempts with the data, evaluate its success, and devise the plan for the next attempts. Although I thoroughly enjoyed the process, everything I tried in that year from Lyupanov exponents to fast Fourier transformations failed to achieve a viable analytical solution. That September, however, the germinating seed broke the surface of the soil.

That September, I first met Isabel Granic after the first session of a seminar course taught by Holly Arrow on complex and dynamic systems. She had just arrived for her post-doc with Tom Dishion and was hoping I could help her with access to the observational data at OSLC. When we met in my office later that afternoon, Isabel gave me something that changed my life: a copy of her doctoral dissertation. She was the first to use state space grids for parent–child interactions and, as soon as I saw it, I knew, this was it. This was what we were looking for in our meetings at OSLC. This was the realization of the vague idea I had over a decade earlier. It captured the complexities of human interaction with a simple elegance. It opened a world of possibilities. Isabel phoned her husband Marc Lewis, the person who first developed state space grids on sabbatical at the University of Oregon, and we met at a pub to discuss state space grids. We closed the place 8 hours later and had embarked on what has become a rich collaboration and friendship that continues to this day.

Within the year, we submitted five state space grid manuscripts together. The following year, I began my doctoral training in Toronto under Marc's supervision. My methodological interests, combined with Marc Lewis' shift into neuroscience, made me the de facto inheritor of the state space grid methodology. Over the next few years, I worked with Alex Lamey—the brilliant programmer who first developed grids with Marc Lewis—to develop GridWare. We launched the website www.statespacegrids.org in 2004 to distribute GridWare for free.

Since that time, I have continued to develop GridWare and the state space grid technique. In 2009, with the help of student programmers at Queen's University (Shawn Drape, Ji Cho, and Vishnu Nair), I launched a new website with an upgraded version of GridWare (version 1.15a with entropy and transitional propensity measures) and the new GridWare File Converter. This past year, I upgraded the web site again to include a user forum to facilitate the growing network of state space grid users around the world.

Over the past decade, I have conducted dozens of state space grid workshops in North and South America and Europe. This has been a fantastic opportunity for me to see a range of data formats and research questions. These experienced have pushed me toward more creative solutions and a better understanding of the range of possibilities that state space grid analysis holds. I am deeply indebted to the hundreds of people in these workshops and those whom I have helped on line. Each vexing problem, odd data format, unique research question, and technical issue has helped in some way to the creation of this book. In a way, this book reads like a 4-day workshop on state space grids.

The objective of this book is to provide the reader with a comprehensive understanding of state space grids and how to conduct state space grid analysis. The book is organized so that each chapter provides the necessary information for comprehending the next chapter. It begins with a comprehensive description of dynamic systems and the foundational concepts from which state space grids were derived in Chap. 1. It is important to note that adopting a dynamic systems approach is not necessary to be able to use state space grids. Often users just want to explore their dynamic data or test more direct hypotheses unrelated to systems concepts. This is perfectly reasonable—we purveyors of dynamic systems are not necessarily a dogmatic lot. Still, I think that one of the great benefits of this technique is to facilitate *thinking systemically*. As described in Chap. 1, there are deep and profound implications for how we think of causation, especially in terms of development. So, you may start out using state space grids for very pragmatic reasons, but the act of working with this technique may in fact inspire greater resonance with systems thinking.

Chapter 2 is a conceptual description of state space grids culminating in the review of all state space grid studies to date. I also include examples from currently unpublished projects to provide as wide a range of project ideas as possible. In the next chapter, I introduce GridWare, the software for creating state space grids. This is a free Java-based program I distribute via the internet at www.statespacegrids.org. This is followed by Chap. 4 which provides the nitty-gritty details of how to create a project in GridWare for state space grid analysis. Most of these issues have to do with the formatting of the data files and basic considerations of data and variable types. The remaining chapters cover the analyses that can be conducted using state space grids, starting with the most basic in Chap. 5, between-grid analyses in Chap. 6, and culminating with advanced analyses in Chap. 7.

As with my workshops, there is a cyclic redundancy across chapters. Concepts first introduced abstractly are then fleshed out in greater detail and eventually become the focal point for analyses. By the end of the book, and if you try the examples provided here, you will be an expert in state space grid analysis. The book should also continue to serve as a reference resource for any further analytical circumstances.

Happy gridding!

Melbourne, Australia
February 2012

Tom Hollenstein

Contents

About the Author

Dr. Tom Hollenstein is the leading expert on state space grids and heads the project to further develop software for state space grid analysis, GridWare (www.statespacegrids.org). He was trained in observational data analysis at the Oregon Social Learning Center and received his doctoral training at the University of Toronto under the supervision of Dr. Marc Lewis, the originator of the state space grid method. Dr. Hollenstein is currently an Associate Professor of Psychology at Queen's University in Kingston, Ontario.

Chapter 1
Dynamic Systems

The complexities of natural systems have plagued scientists and philosophers for centuries. Even as Newton was tackling and solving all sorts of problems of celestial mechanics, he and his colleagues struggled with what would seem to be a minor variation on the motions of the planets. With his knowledge of gravity, Newton could predict the orbital trajectories of two planets on the basis of their density, distance, and speed. Add a third planetary body and the predictive accuracy goes out the window. Though it is difficult to pinpoint an exact moment when a scientific endeavor was born, this comes close to being the birth of systems science. As there were myriad basic laws and relations to be discovered through linear and mechanistic formulations, the nonlinear dynamics of complex systems were not, however, the object of inquiry for many years.

Flash forward to the middle of the twentieth century. The advances of electronic computing combined with the growing sophistication of all the sciences led to the inevitable need for a more comprehensive account of complex interrelations. As always, this push was led by physics and mathematics, but soon included the natural sciences, computer science, and, eventually, social science. Pioneers such as Norbert Weiner and Ross Ashby developed the field of cybernetics as the science of information and control in machines and natural organisms (Ashby 1947; Weiner 1950). The capacity for machines to self-regulate led to the understanding of complex mechanisms in biological organisms. Chief among these were feedback process and the circularity of *self-organization*—the ability of systems to "interact with themselves and produce themselves from themselves" (American Society for Cybernetics). From cybernetics emerged General System Theory (GST), most notably advanced by von Bertalanffy (1968). With the accumulation of observations and the burgeoning understanding of certain lawful regularities in complex systems, there was a movement to identify those processes which were generalizable to all systems. According to von Bertalanffy, GST was an attempt to provide "models, principles, and laws that apply to generalized systems or their subclasses, irrespective of their particular kind, the nature of their component elements, and the relationships or 'forces' between them" (von Bertalanffy 1968, p. 32). Thus, the bold claim of this theory was that the complex interactions through which celestial bodies formed planetary systems or galaxies, the complex interactions among atoms to

T. Hollenstein, *State Space Grids,* DOI 10.1007/978-1-4614-5007-8_1,

form molecules, and the complex interactions of system elements at every scale in between were all connected through the same lawful processes of self-organization. A grand unified theory indeed!

The next steps were to identify which systems were applicable and which processes were generalizable. The most basic distinction of a system is whether it is "open" or "closed." An open system is one in which energy is exchanged with the external environment. In contrast, a closed system does not exchange energy with the environment and, therefore, does not function with the same processes as an open system. A basic example of a closed system is a machine with a single function. The best example of an open system is a living organism. Thus, biologists such as von Bertalanffy began drawing on the advanced understanding that physicists and mathematicians had derived from highly controlled measurements for describing the behavior of humans and other organisms (Box 1.1).

Box 1.1 Dynamic or Dynamical: What's the Difference? There is some confusion about terminology that naturally arises with such an interdisciplinary approach. A system, by definition, is a collection of elements that change over time (Thelen and Smith 2006), so it is necessarily dynamic. However, it is not necessarily "complex."[1] The situation is further obfuscated by the use of both "dynamic" and "dynamical" as adjectives describing systems. Some of these differences in terminology are meaningful, in which they distinguish salient features of the system, and others are simply different disciplinary approaches. Thus, it is useful to clarify these so that you, the reader, can better understand systems research and theory.

For mathematicians and some other systems researchers, the equations that describe complex behavior *are* the system. That is, the system is an exhaustive, quantifiable set of relations among variables that are iteratively determined over time (i.e., the condition of the system at time t is derived from the condition at time $t-1$). From this viewpoint, systems do not exist without this quantification. When the term "dynamical systems" is used, it most often reflects this approach.

In contrast, there are those who begin with the premise that natural systems exist, whether or not we quantify them. These are open systems with varying degrees of quantifiability. In the human domain which is our concern here, there are some processes that are easily quantified (e.g., body movements) and others less so (e.g., anger). Still, for those who tend to employ the term "dynamic systems," there is the fundamental assumption that individuals or groups of individuals function as open, complex, and adaptive systems. The trick is to discover the means by which the basic properties of dynamic systems operate in human behavior.

[1] The term "complexity" has a very precise meaning in mathematics and physics. It is beyond the scope of the present treatment to review complexity science. The reader is encouraged to refer to excellent works by Kauffman (1995); Cohen and Stewart (1994).

The theoretical framework that forms the basis of state space grids and my own work is that of the "dynamic systems" (DS) approach (see Box 1.2). The basic premise is that since humans are biological organisms, they are also DS. From there we can set about trying to discover how. That being said, I do not think this is in direct contrast with the more mathematical approaches of "dynamical systems." With sufficient measurement (a challenge in developmental psychology to be sure, see Chap. 3), the precision and theory testing made available by differential and difference equations, and other modeling techniques is essential to the advancement of our knowledge. Some of these will be discussed in Chap. 7. Where I and others draw the line is to think of these formulae, functions, equations, and models as anything but representative of the actual systems we are striving to understand.

Features of Dynamic Systems

The central feature of a complex dynamic system—perhaps most difficult to comprehend—is that it comes into being via *self-organization*. With self-organization, the interactions among lower-order elements actually give rise to the superordinate structure of the system (e.g., tissue is an emergent structural organization from the interactions among cells). This process, sometimes called emergence, is at the heart of any DS approach (e.g., Haken 1977; Lewis 2000a), even if never explicitly stated. It is easy to see why self-organization has had particular appeal for developmental scientists (Lewis 2011; Thelen and Ulrich 1991; Witherington 2007), who by definition grapple with the problem of the emergence of novel forms of behavior across the lifespan.

Self-organization is a bottom-up explanatory mechanism of development with extraordinary implications. The physicist who won a Nobel Prize for his work on self-organization in thermodynamics, Ilya Prigogine, has boldly declared that the discovery of self-organization has meant the end of determinism (Prigogine 1997). At best, he argues, we can obtain statistical knowledge of self-organizing phenomena, but never predetermine the exact form or trajectory. Others have argued that self-organization renders reductionism obsolete as a scientific approach (Thompson 2007) for the same reason. This is not to say that the behavior of self-organizing systems is indeterminate, rather it is not reducible to efficient cause in the Aristotelian sense (Witherington 2007). At the end of the day, that is the massively important contribution of self-organization—a shift in how we think about causation in naturally occurring DS.

To illustrate these causal dynamics and explain the process of self-organization vis-à-vis development, consider the model depicted in Fig. 1.1. Through this illustration, the hierarchical structures that emerge through self-organization can be explained in terms of bottom-up and top-down causal processes. Following this

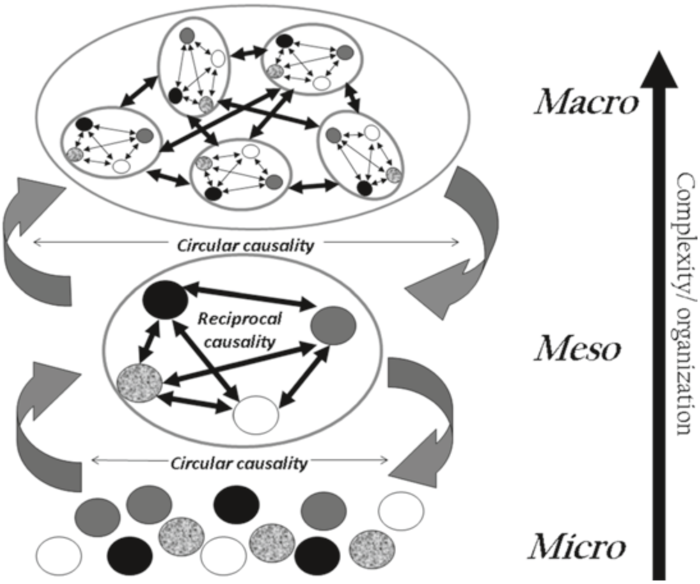

Fig. 1.1 An abstract illustration of the organization at three levels of a dynamic system. (Adapted from Hollenstein 2012)

explanation, several key characteristics of DS will be covered that have the most direct relevance for state space grids.

The first thing to notice about the model depicted in Fig. 1.1 is that it is segmented into three levels of organization or complexity to depict the part–whole relations common to all systems theories. Thus, the system is depicted as a nested hierarchy with the least complex, most rudimentary components at the bottom level, here labeled as "micro," and the most complex and advanced components at the top level, here labeled as "macro." In between these two levels is an intermediate one, here labeled as "meso," that is of a higher order than the micro scale but of a lower order than the macro scale. Although three scales are depicted, there are further lower order scales below micro and higher order scales above macro (not shown). The meso scale is typically the focus or unit of analysis with the understanding that it is made up of the elements from the micro level and simultaneously functions as the elements for the macro level (Hollenstein 2012).

If the description were to stop here, leaving only a nested hierarchy, it would be nondynamic yet similar to some other "systems" models. That is, though the idea of nested interrelations is not radical or uncommon, often these models are static. This fixation on the categories or states or elements of such models is static because it neglects *how* interactions within and between levels transpire and explain system behavior. In other words, for this type of modeling there are many nouns but few verbs.

Instead, with the dynamics at the fore, the focus is on time and change. This is the stuff of development, the concern of embryologists and epidemiologists, economists and therapists, parents and educators, historians and politicians. How do novel forms emerge? How do they grow and change over time? How do they stabilize and, once stabilized, undergo transformation? Through the process of self-organization, time and hierarchical organization are inexorably linked. Increases in complex organization occur across both developmental and evolutionary time scales (indeed this is the link between ontogeny and phylogeny, see Lewis and Liu 2011). Humans, and the societies that they build, become increasingly complex over these time scales. For this reason, the structural relations of organization scales depicted in Fig. 1.1—micro, meso, and macro—are also relations of time scales.

To make this more concrete and to put in terms that relate most directly with issues of human development, we can identify the time scales in Fig. 1.1 as real-time, moment-to-moment processes (micro), the situational, day-time processes that occur over minutes to hours (meso), and the developmental-time processes that unfold over the course of months or years (macro). In the bottom-up process of self-organization, the interactions among the micro elements that transpire in real time create structures that persist and recur, thus forming the elements at the next higher meso level. These stable and recurring structures at the meso level then build the habits and identities of the individual across development. For example, Marc D. Lewis has modeled emotional development in this way (Lewis 2000b). Emotions are made up of feeling states and appraisals that cohere in the moment and are most often fleeting occurrences. However, they can sometimes persist and become more stable over the meso time scale of minutes or hours in the form of moods. Developmentally, the emotional experiences of childhood all occur in real time, yet some emotional states and appraisals recur with a frequency greater than others due to myriad factors such as temperament, family socialization practices, and the stress level of the home environment. This is most obviously underpinned by biological substrates—the emotional experiences that persist into moods become strengthened and literally build neuronal structures through neural plasticity (Lewis 2005). Over the course of development, these emotion structures that recur with the greatest frequency and intensity form the basis of what is known as the personality—the aggregate product of all previous activities at the micro and meso scales.

This bottom-up emergence is only half of the causal picture, however. The progression of development is also a product of system-generated top-down processes. Within a particular time scale, interactions among elements are reciprocally causal—they mutually influence each other as a matter of their interactions. From these reciprocal interactions among lower order elements, a higher order structure emerges. Yet at the same time, these higher order structures constrain the interactions among lower order elements in a top-down fashion. Consider the example of emotions and moods from the Lewis (2000, 2005) model. The coordination of cognitive appraisals and internal feeling states in an anger episode at the micro scale may transform into a mood and persist for the next several hours. During this irritable mood, the recurrence of the real-time emotional state of anger is much more probable than during a neutral or happy mood. In this way, the meso-scale mood constrains the micro-scale emotions. The combination of these bottom-up and top-

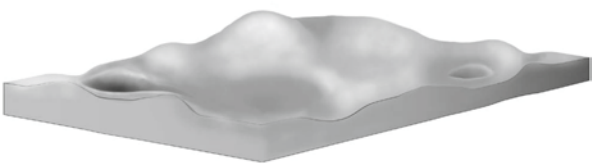

Fig. 1.2 Hypothetical state space

down causal processes characterizes the relations between time scales and is called circular causality (Haken 1977).

To summarize, the structure of a complex, open and dynamic system is hierarchically nested and the result of reciprocally causal relations within scales and circularly causal relations between scales. This is the self-organizing process of development and the overarching framework of DS models. Given this background, we can now consider some of the key characteristics of system behavior that arise from these processes: multistability, attractors, repellors, and phase transitions.

Multistability and Attractors

Although self-organizing systems have many possible states, they generally only have a few stable states that recur with some regularity. These recurrent states are called attractors because they "pull" the trajectory of the system toward those states. In contrast, some states of the system rarely, if ever, occur. These are called repellors. The configuration of a system then can be described in terms of the underlying set of attractors and repellors revealed by its behavior. To illustrate this multistable configuration, DS models are often represented on a state space, an abstract space which contains all possible states of the system. As a heuristic, a state space is often depicted as an undulating landscape (see Fig. 1.2). The current state of the system can be thought of as a ball rolling across the landscape, easily getting caught in the wells or holes that are attractors, and never coming to rest atop the hills that are repellors. As we are talking about open systems, there is a flow through of energy and it is this energy that is required to move the system through its sequence of states. So, for an attractor, it takes very little energy to enter and substantially more to get out. This very basic distinction of some states being significantly more probable than others has become one of the most direct ways to examine developmental phenomena. As we discuss later, this multistable nature of systems has been examined by quantifying attractor states in motor coordination (Kelso 1995; Thelen and Ulrich 1991), psychopathology (Johnson and Nowak 2002), emotional states (Lewis et al. 1999), and most often in interpersonal relationships (Fogel 1993; Fogel and Thelen 1987; Gottman et al. 2002; Granic and Lamey 2002; Hsu and Fogel 2001; Licktwarck-Aschoff et al. 2008).

Given that a system can be conceptualized as a configuration of attractors and repellors on a state space, variability is the essential means of information about a system (Fogel 1993; Granic and Hollenstein 2003, 2006; Thelen and Ulrich 1991;

van Geert and van Dijk 2002). In most ways, the term "dynamic" in DS refers explicitly to system variability or changes over time. What the variability of system behavior reveals then is the underlying structure—the state space configured by attractors and repellors. Consider two contrasting systems as an example. In the first, there are several attractors, each of low-to-moderate "strength" (e.g., the depth and width of the broad basin in Fig. 1.2); in the other system there is only one very strong (i.e., deep) attractor. The behavior of the first system would be much more variable as it would take minimal energy to move in and out of the attractors on the state space. In contrast, the second system would show little variability as most of the time would be spent at the bottom of the one large attractor. This relationship between variability and attractors highlights an important distinction that satisfies von Bertalanffy's (1968) GST mandate: content versus structure. Content is the specific states of a system. As described earlier, each system has the theoretical potential to be in any one of the states on a state space, yet only a few out of that exhaustive set tend to occur and recur. Why? Because of the structural configuration of the state space. Thus, content can only be understood in terms of the structure that constrains which content occurs and in what way.

For example, an interpersonal system like a married couple might have several specific joint emotional states in their behavioral repertoire, but it is the temporal pattern of those structurally constrained states that reveals the nature of the couple's relationship (e.g., Gottman et al. 2002). That couples fight is not distinctive; that they get stuck in conflict for long periods of time is.

Phase Transitions

Up to this point, I have described systems in terms of stable configurations in order to illustrate multistability, attractors, repellors, and variability. However, systems are also adaptive and evolving, and therefore we also need to understand how they change and grow. Although the variability described above is dynamic change, a system could continue for a very long time with the same stable structure giving rise to that dynamic variability. This would be an example of no change or growth. When systems do change, it is at the structural level and comes as a qualitative shift in system dynamics. In DS terms, these qualitative shifts are called phase transitions. Phase transitions are characterized by a range of changes in system dynamics (Guastello 1995; Hartelman et al. 1998; Thelen and Ulrich 1991), most notably a temporary increase in variability as old structures break down to give way to new ones. Hence, both prior to and after a phase transition, the system is relatively stable with lower variability and greater predictability. Importantly, the patterns of system behavior are qualitatively different before and after a phase transition, by dint of differing attractor configurations on their respective state space.

This pattern of low-high-low variability is the most basic form of a phase transition and one that is easily detectable in a wide range of circumstances. In particular, qualitative shifts in system behavior are characteristic of the stage-like progression of development similar to that of a growing child that achieves distinctively more complex ways of integrating input and coordinating output (Piaget 1952). This has led to the developmental phase transition hypothesis that at key transition points between developmental stages there will be corresponding increase in variability (Granic et al. 2003, Hollenstein 2007, 2011b; Lewis et al. 2004). For example, the onset of adolescence is a drastic reorganization in the biological, cognitive, emotional, and social domains. In systems terms, this is not just change in the variability of behavioral content but a structural reconfiguration of the state space. In the most direct test of the adolescent phase transition hypothesis to date, family interactions were shown to temporarily increase in variability during the transition into adolescence and then return to previous levels of variability shortly after (Granic et al. 2003).

In summary, there are key features common to all DS that have been discovered. For our current purposes, those most notable are the structural properties revealed by attractors, repellors, and state space; the dynamics revealed by variability; and the phase transition as the mechanism of change. Over the past 20 years, there has been great progress in applying these DS constructs to developmental phenomena (Hollenstein 2011a); yet most of this promise has been realized in the theoretical rather than empirical domains (Granic and Hollenstein 2006). The empirical bottleneck is mostly due to a dearth of methods with which to test the rich theoretical models. Hence, the primary purpose of state space grids and this book is to provide an accessible and flexible methodological tool for a wide range of researchers to be able to visualize and measure variability, attractors, and phase transitions.

Box 1.2 Dynamic Systems Theory Is "dynamic systems" (DS) an approach, a perspective, a model, a theory, a metatheory, or a framework? As it derives from General Systems Theory, there is a tendency to tag on "theory" after the term "dynamic system" as well. Yet because of the scope of generalization, it goes beyond the specifics of theoretical modeling in any particular domain. As such, it is unlike what is typically meant by a theory which makes specific predictions for specific phenomena under specific conditions. Further, DS "theory" is not necessarily contradictory to these domain-specific theories either. How then shall we model development as a dynamic system?

First, it should be noted that the acronym DS is an adjective used to describe an approach, model, etc. Second, the most circumspect utilization of systems principles would best be described by a DS "approach" or the DS "perspective." In this way, the concepts of variability, phase transitions, attractors, and repellors are applied to developmental phenomena less formally than a theoretical model would imply. A DS approach to depression, for example, would be to examine depressive episodes as cyclical attractors. This research question would be relatively independent of existing theories of depression (it would be based on the observed phenomena of depression shared by all theoretical approaches to the disorder). It would also be relatively independent of a more formal test of individuals as DS. Typically, the DS approach is empirically or methodologically driven in domains where formal theorizing is not yet possible due to insufficient data or is less important.

Third, at the computational or statistical level, DS modeling is yet another terminological distinction. Here a prescribed pattern of relations among system elements, usually articulated as an equation, is tested for its fit to actual or simulated data. Catastrophe models, agent-based modeling, and coupled equation models fall into this category. These models may or may not necessarily be a test more formally of characteristics of open DS.

Finally, the leading proponents of DS theorizing in developmental science view DS as a metatheory or a metatheoretical framework (Granic and Hollenstein 2003, 2006; Lewis 2005; Spencer et al. 2006; Thelen and Smith 2006; Witherington 2007). "Theories and methods refer directly to the empirical world, while metatheories refer to the theories and methods themselves" (Overton 2007, p. 154). Hence, this approach does not deny the validity of domain-specific theories but instead contextualizes them with a set of processes that explain the phenomenon at hand.

Conclusion

From this review of DS, I would like to highlight a few important points that may also aid in understanding of the rest of this book.

1. The natural world is a set of complex, adaptive, open, and self-organizing DS. As part of that world, human beings, both intrapersonally and interpersonally, are DS as well. That is, they function according to the same processes of all DS.
2. To understand human development, it is necessary to recognize the nested, hierarchical temporal structure of human systems. At the very least this impels researchers to view their phenomenon of interest at the meso scale—an emergent set of structural relations from the micro scale constrained by the structures at the macro scale.
3. The DS perspective is a radically different approach to causality. Therefore, it is not merely the application of different methodologies that will provide the insights into human development that we seek. Instead, understanding the complex nature of human behavior requires a challenge to assumptions of both linear and efficient causality.
4. Any scientific endeavor first begins with observation and description. Given our relatively nascent understanding of the human being as a dynamic system (and the challenges to determinism that implies), the first priority is to obtain the means for providing that description. The state space grid method is one technique that is uniquely poised to fill that need.

The focus of this book is the state space grid method. I have provided a sufficient review here to be able to understand the method and the analyses conducted to test DS-based hypotheses. However, the DS approach to development is a deep and broad undertaking. There are many excellent books and reviews that I strongly recommend to the reader interested in pursuing these ideas further. Esther Thelen was a superb writer and thinker and I would highly recommend all of her works. Most notable among these would be her monograph on child development (Thelen and Ulrich 1991), the two-volume book written and coedited with Linda Smith (Thelen and Smith 1994), and a recently updated handbook chapter (Thelen and Smith 2006). Paul van Geert has also provided a great deal of clear explanations and methodological techniques in books (van Geert 1994) and numerous articles (Steenbeck and van Geert 2008; van Geert and Steenbeck 2005; van Geert and van Dijk 2002; van Geert 1998). There are a couple of special issues in journals that also provide the breadth of approaches: The British Journal of Developmental Psychology (Vol. 16, 1998) and Child Development Perspectives (Vol. 5, 2011). Marc Lewis' comprehensive modeling of socioemotional development (Lewis 2000a), the neurobiological dynamics of emotion (Lewis 2005), and the relations across real-, developmental-, and evolutionary-time scales (Lewis and Liu 2011) is exceptional in both its coherence and impact. Finally, Isabel Granic has adeptly and articulately reviewed DS methodology (Granic and Hollenstein 2003, 2006) as well as provided detailed explanatory models of DS mechanisms in the development of psychopathology (Granic 2000, 2005; Granic and Patterson 2006). There are, of course, many other high-quality authors and sources but, at the time of this writing, these are the sources I would recommend starting with.

Chapter 2
State Space Grids

Now that some of the dynamic systems (DS) terminology is familiar to you, we can see how the concepts of state space, attractors, repellors, and phase transitions can be applied with state space grids. In this chapter, I will describe how state space grids were derived from the abstraction of state space. Next, I will describe the essential features of state space grids. In the last section, I will review state space grid studies to date, so that you may get a sense of the versatility of the technique.

From Continuous to Categorical

State space is often depicted in its most simple form as a two-dimensional plane formed by the intersection of two perpendicular dimensions or axes. There is no theoretical limit to the number of orthogonal dimensions necessary to describe a system on a state space, but after the third dimension it becomes difficult to visualize. Since there is nothing ostensibly different about higher dimensional state spaces, the utilization of two dimensions is sufficient for illustrative and analytical purposes (see Chap. 7 for an expansion beyond two-dimensional state space).

Consider the state space landscape depicted in Fig. 2.1. Here, each position on the landscape is a possible state in which the system could *hypothetically* be at any one time. Furthermore, each location is a combination of one value along the *x*-dimension and another value along the *y*-dimension (let us leave the *z*-dimension out for the moment). Thus, it is possible to identify each point on the landscape in terms of an *x* and a *y* coordinate, similar to a map. At the risk of being overly simplistic, there are at least four important considerations of such an arrangement. First, the values on any one of those dimensions must be mutually exclusive. If, for example, the *y*-dimension ranged from 0 at the bottom-right corner to 100 at the top-right corner in Fig. 2.1, then at any given time point there could only be one value to represent the system on that dimension. That is, *y* could only be at value 15 or 34, not 15 *and* 34. The same is true for the other dimensions. This may seem obvious but, as you will see later, mutual exclusivity is a necessary condition for state space grids.

T. Hollenstein, *State Space Grids,* DOI 10.1007/978-1-4614-5007-8_2,

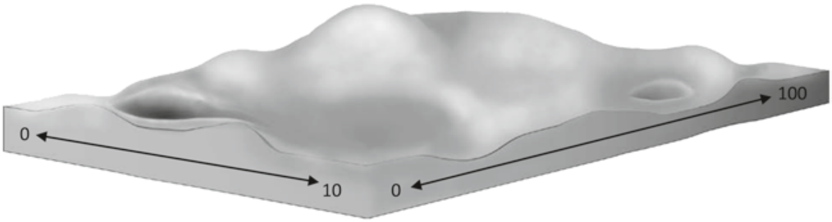

Fig. 2.1 Hypothetical state space with ordinal dimensions

Second, the range of values along each dimension must be exhaustive in which there are no other possible values that could ever occur. Using the previous 0–100 example, this could be exhaustive if the dimension represented a percentage wherein values below 0 or above 100 were simply not possible. If each dimension conforms to this requirement, then the state space will also by definition be exhaustive as well. Again, this may seem obvious or overly simplistic, but it will be important later.

Third, although most often the state space is depicted as square, the scale and/or range of each dimension does not have to be equivalent. The y-dimension could be 0–100 while the x-dimension could be a range of 0–10. Of course, with continuous scales (i.e., ratio scaling), the number of intervals between values is a function of the level of precision of the measure or index. If the level of precision is one unit for the y-dimension (0–100) and 1/10 of a unit for the x-dimension (0–10), then both the dimensions would have 100 scale units between minimum and maximum values. Furthermore, the distance between those values depicted on the state space diagram can be arbitrarily determined—they may be constrained to equivalent lengths to make a square state space or allowed to maintain the same scale units to make a rectangular (10×100) state space. As we will see in Chap. 7, there are ways to relate continuous measures to categorical dimensions. For now, we will proceed by considering categorical dimensions.

The final consideration stems from the third: the state space can be derived from ordinal, categorical, or nominal dimensions, as long as each dimension is comprised of mutually exclusive and exhaustive values. For example, if the dimension of interest was an index of emotional valence—ranging from negative to positive—the degree of continuity of the values on that dimension would be a function of the measurement precision. Valence could be captured by a self-report joystick that recorded values ranging from 0 (negative) to 100 (positive), or a visual analogue scale ranging from 0 to 10, or from the real-time ratings of a trained observer in only three categories (negative, neutral, positive). All three measurement methods would be appropriate as a dimension of state space; the only difference would be in the number of scale intervals or categories (100, 10, or 3). Two points can be taken from this example. One is that beyond the limits of the precision of the measurement all dimensional variables become categorical—a measurement unit of miles seems continuous from space but is categorical when looking at a 4 square mile area (as-

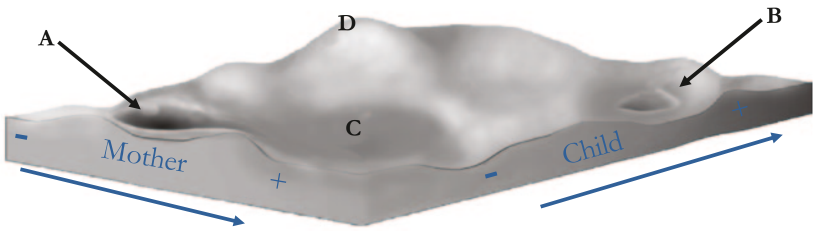

Fig. 2.2 Hypothetical state space of a parent–child system. *A* Deep (strong) attractor: Mother Negative with Child Negative (“Mutual Negative”); *B* Shallow (weak) attractor: Mother Positive with Child Positive (“Mutual Positive”); *C* Attractor basin: Mother Positive with Child Negative (“Permissive Parenting”); *D* Repellor: Mother Negative with Child Positive (“Harsh Parenting”)

suming no further break down in units). The other is that each dimension of the state space can be broken down into categories, similar to the A–Z and 1–10 format of road maps, to create a grid of intersecting categories. Now, with such a categorical approach to the dimensions of the state space, then, the range no longer needs to be an ordinal or quasiordinal sequence, but can also be in nominal categories with no numerical values or ranking to determine their order or adjacency. In other words, to analyze the state space of a system, we can overlay a grid of meaningful boundaries to discriminate distinct x/y states. This is the basis of state space grids.

Before describing state space grids, let us consider a real-world example of the state space depicted in Fig. 2.1. Often the system being analyzed by state space grids has been the parent–child system, with individual states for each dyad member ranging in valence from negative to positive. Superimposing that system on the hypothetical landscape, we can get something similar to the state space depicted in Fig. 2.2. Here, Mother behavior is depicted from negative to positive along the x-dimension and Child behavior is depicted from negative to positive on the y-dimension. Now, the configuration of valleys and peaks representing attractors and repellors, respectively, can correspond to the structure that underlies this particular dyadic system. The dyad in this example struggles with problematic behavior. The strongest attractors are in the mutual negative region indicating that when they get into that state it is difficult for them to get out of it. Fortunately, they are also “pulled” into a mutual positive state at times, though not as strongly as the mutual negative state. Although not a strong influence in the Child negative with Mother positive basin, the dyad covers a broad area that could be labeled permissive parenting. Finally, as seen from the repellors “hill,” this parent is never harsh or negative while the child is positive. In this way, the state space landscape can provide a static picture to depict the structure and the range of system behavior. The state space grid method goes a step further by taking this general idea and using it to analyze the temporal dynamics—the sequence of system states as they progress through time—in order to derive the structure and function of these systems.

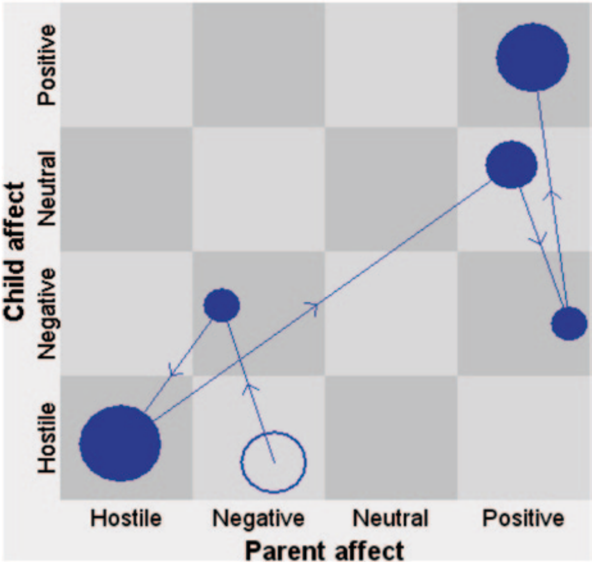

Fig. 2.3 A hypothetical state space grid example of a mother–child interaction sequence

State Space Grids

State space grids are two-dimensional state spaces with at least two mutually exclusive and exhaustive categories on each dimension. The intersection of these categories forms a grid of cells representing each categorical combination. The number of categories on each dimension does not need to be the same, so the state space grid can be any size rectangle. The cells themselves do not need to be square either, though square cells are perhaps the clearest way to present a grid and this is the manner in which all state space grids have been presented to date. Remember that the state space is the space of all possible states so that what must be in that space are literally ALL the possibilities, theoretical and actual. See Chap. 4 for further discussion of this issue.

Most often, the blank state space is generic in that the combination of states could correspond to any number of systems of the same type. Some systems might spend most of their time in one region while other systems spend their time in other regions, with little or no overlap. For example, as we shall see later in this chapter, many of the published reports of state space grids focus on the parent–child system. In these studies, the state space grid remains constant (e.g., Parent negative, neutral, positive with Child negative, neutral, positive) but each parent–child dyad behaves differently. That is, their sequence of states is unique on a generic state space. Thus, it is the plotting of this sequence of states—the *trajectory*—on the state space that reveals the structure of any particular system (Fig. 2.3).

The trajectories are plotted on a state space grid as nodes or plot points connected by lines to indicate the transition from one state to the next. Thus, the sequence of states is a series of *visits* to cells on the grid. When information about the duration of each state is available, the size of the node corresponds to the duration of that state. In Fig. 2.3, we see a short trajectory of behavior on a state space grid that is similar to an aerial view of the state space landscape in Fig. 2.2. With four categories of affective valence for both mother and child, the behavior depicted for this dyad begins in their mutually negative attractor (the open node depicts the starting point) but then they resolve into mutual positivity.

A wide range of measures can be derived from trajectories plotted on state space grids in this way. These measures can tap both the content (i.e., the frequency and duration of specific states) as well as the structure (i.e., the configuration of attractors and repellors) by quantifying the dynamics. All of these measures are described in great detail in subsequent chapters.

State space grids are a flexible analytical tool, which can be useful in a wide array of research circumstances. This book will highlight many of these possibilities but as many or more of these applications of the method are yet on the horizon. In general, though, for any user there are three potential uses of state space grids. First, state space grids may simply be useful as a tool for visual inspection of data. I happen to ascribe to John Tukey's notion of exploratory data analysis (Tukey 1977) as an essential, though often neglected, research function. Indeed, in any new domain of inquiry, the first order of business is description. If that is all state space grid analysis achieves, it would be a great contribution to science.

Second, by extension, the DS approach in general and the exploratory data analyses facilitated by state space grids in particular are a rich source of hypotheses about behavioral *processes.* As I presented in Chap. 1, DS approaches require a different way of thinking about intrapersonal and interpersonal behavior, time, and causality. Though there is growing theoretical interest in the processes and mechanisms underlying behavior, especially in the developmental sciences, there is a paucity of research tools with which to test these theoretical claims (Granic and Hollenstein 2006; Richters 1997). Thus, in the state space grid method lies the potential generation of testable hypotheses and the enactment of both inductive and deductive scientific means.

Finally, state space grids are a source of variables and analyses of both structure and content. These measures and analyses can then be combined with or compared to existing, more traditional methodologies to test various content-specific and structural hypotheses. As it is a metatheoretical framework (Witherington 2007), DS techniques such as state space grid analysis can be incorporated into existing theoretical or statistical models that are domain specific (e.g., Granic and Patterson 2006). Realistically, the scope and power of state space grids as a dynamic analysis tool has not yet been fully realized and it is up to the current and next generation of researchers to push and extend the technique into the greater prominence it deserves.

A Review of Studies Using State Space Grids

I will briefly review the state space grid studies that have been published as of 2011. This will be followed by a review of the work in progress or otherwise unpublished to give the reader a sense of a wider range of possibilities. Details presented here are necessarily brief, though we will return to some details of these studies in later chapters that focus on analysis. The reader is encouraged to read the original sources identified in Box 2.1.

Box 2.1 State Space Grid Studies to Date

Study	System	Age	Structure	Content	Grid size
Granic et al. (2012)	Mother–child	8–12 years	Variability/ flexibility	Specific affect	9×9
Granic et al. (2003)	Parent–boy	9–18 years	Variability; phase transition	Emotion valence	4×4
Granic and Lamey (2002)	Mother–child	8–12 years	Attractors	Emotion valence	4×4
Granic et al. (2007)	Mother–child	8–12 years	Variability/ flexibility; repair	Specific affect	9×9
Hollenstein and Lewis (2006)	Mother–daughter	11–12 years	Variability/ flexibility	Specific affect	10×10
Hollenstein (2007)	Mother–daughter	10–14 years	Variability; phase transition	Specific affect	10×10
Hollenstein et al. (2004)	Parent–child	4–6 years	Variability/ flexibility	Emotion valence	4×4
Lewis et al. (1999)	Infant	2–6 months	Attractors	Distress and attention	5×5
Lewis et al. (2004)	Toddler	14–24 months	Variability; phase transition	Engagement with mother and frustrating toy	5×5
Lunkenheimer et al. (2011)	Parent–toddler	3–4 years	Variability	Emotion valence	4×4
Martin et al. (2005)	Peer interactions	4–5 years	Attractors	Gender, social competence, valence	2×3×5

The first study to use state space grids was the one for which they were developed (Lewis et al. 1999). Marc Lewis and Alex Lamey devised the method to be able

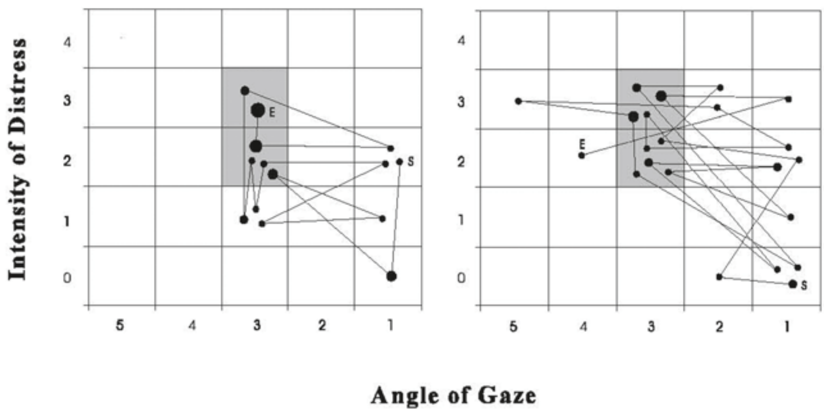

Fig. 2.4 State space grids from Lewis et al. (1999)

to depict and measure the sequence and recurrence of infant socioemotional states corresponding with their simultaneous levels of distress and attention to their own mother who was sitting close by. That is, the infant's emotional state and social behaviors to regulate that emotion could be analyzed as distinct but coordinated processes. Infants were given frustrating toys while playing on the floor next to their mothers. One minute of their level of distress (0–4) and angle of gaze toward mother (1 = gaze averted to 5 = direct attention to mom) were recorded and plotted on state space grids (see Fig. 2.4). The authors examined whether infants' socio-emotional attractors at 2 months of age were stable through 6 months of age.

The Lewis et al. (1999) study introduced attractor analysis using state space grids through a two-step process. First attractors were identified through a winnowing process based on the proportional duration in each cell. For each infant, one or two cells were identified for which proportional durations exceeded chance. Second, they measured the strength of these attractors as *return time*, the duration of intervals between visits to the attractor cells. Shorter return times indicated a stronger attractor. As hypothesized, infants tended to have the same attractors at both 2 and 6 months of age and the correlations between the strength of those attractors over those 4 months range from 0.7 to 0.8. Thus, individual differences in emotional and regulatory habits of infants seem to be well established early in infancy and remain stable for the first half year.

Granic and Lamey (2002) extended the state space grid technique from the focus on an individual as the system to the parent–child dyad. In this study, the authors were interested in distinguishing subtypes of children with externalizing problems, specifically those who were comorbid with internalizing problems (MIXED) and those who were pure externalizers (EXT). These children and their mothers were observed during a 6-minute conflict discussion. As the best way to reveal the nature of system dynamics is through perturbation, a mild perturbation was included in this research design. With 2 minutes left to go, dyads were perturbed by the research assistant who knocked on the door, came into the room, and told the dyad to "wrap up on a good note." Before the perturbation, the MIXED and EXT dyads were indistinguishable. However, after the perturbation, the MIXED dyads shifted to a mutually hostile attractor (the state space grids were 4×4 as in Fig. 2.3) while the EXT dyads remained in the permissive region as these mothers pleaded with affection to get the child to focus on and resolve the conflict. Thus, this study revealed unique system dynamics associated with differential socioemotional experiences of MIXED and EXT children.

Following the ground-breaking Granic and Lamey (2002) study, the next group of studies focused on the parent–child dyad as the system and specifically examined the variability in these interactions. In one subset of these studies, the connection between the dyadic variability and the development of psychopathology was explored (Granic et al. 2012; Granic et al. 2007; Hollenstein et al. 2004; Lunkenheimer et al. 2011). The other set of variability studies tested the developmental phase transition hypothesis: transitions between stable developmental periods would be evidenced by the temporary increase in variability characteristic of a phase transition (Granic et al. 2003; Hollenstein 2007; Lewis et al. 2004). Before reviewing these two sets of studies, it is important to first consider the psychological meaning of real-time emotional or affective variability.

Variability and Flexibility

As mentioned in Chap. 1, variability is considered vital information about a system. Yet, variability is a broad term with many viable ways to define or operationalize it. It can be taxonomic, simply denoting that a range of elements exist in a particular domain (e.g., eye color). It can also be used statistically, in terms of deviations from a fixed value (e.g., standard deviations). Variability can refer to variations in temporally contiguous events (e.g., walking stride) or a range of consistency across sporadic events (e.g., hit-or-miss food quality at a restaurant). Of these different meanings of variability, some can be interpreted as flexibility, or the process by which a state transforms into a different state in an adaptive response. For developmental DS applications, most often we have been concerned with variability at two scales.

At the first scale, there are the moment-to-moment fluctuations of system behavior that vary because of the reciprocal influence of the elements of the system (micro scale in Fig. 1.1). For example, try standing on one leg for a few minutes. You will notice that in order to maintain the stance, you will engage in many small muscle contractions throughout your body, mostly in your feet. Let us call this kind of moment-to-moment change *dynamic variability*. At the second level or scale are the variations in system behavior that come from larger, typically exogenous, perturbations. Variability at this scale reflects the degree to which a system can adapt to the changes in the environment. To continue the example, notice what you do differently when someone pushes you while you stand on one leg—different muscle groups become involved. Perhaps you need to swing your arms or bend at the

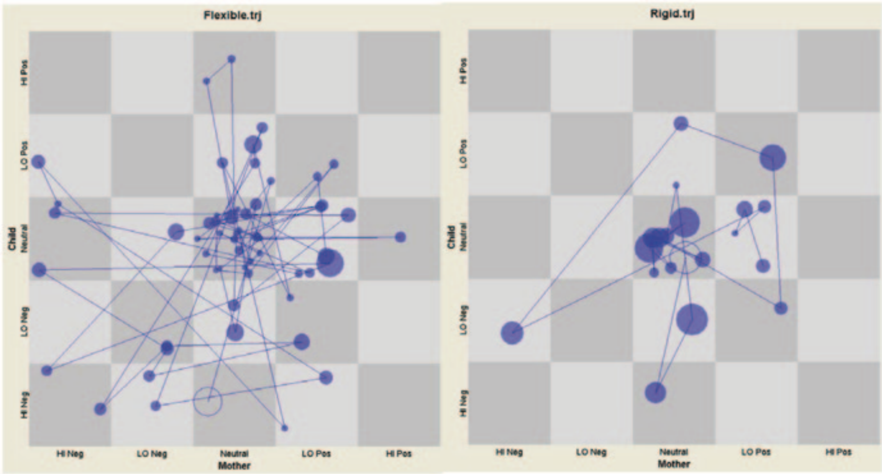

Fig. 2.5 Examples of flexible and rigid interactions on state space grids. The grid on the *left* depicts a more flexible interaction by dint of greater dispersion across the state space, more transitions, and lower mean durations (*smaller nodes*). In contrast, the grid on the *right* depicts a relatively less flexible interaction due to less dispersion, fewer transitions, and longer mean durations

waist. Let us call this adjustment to environmental demands *reactive variability*. In both cases, not dynamically adjusting or not reacting to contextual changes would characterize a rigid system (standing rigidly assures falling over). For dynamic systems in the natural world, the "hallmark of successful adaptation is flexibility in the face of changing physiological and environmental demands" (Thayer et al. 2012, p. 748). Thus, some aspects of variability can be interpreted as *dynamic flexibility* and others as *reactive flexibility* (Hollenstein et al. in press).

With state space grids, dynamic flexibility can be examined through within-grid analyses (Chap. 5) and reactive flexibility with between-grid analyses (Chap. 6). Consider the examples of dyadic emotional valence in Fig. 2.5. Immediate visual inspection of these two trajectories typically leads to the conclusion that the one on the left is more flexible than the one on the right. State space grid analysis offers the means to quantify those qualities with measures such as dispersion (range of cells occupied controlling for relative durations), transitions, and average mean duration across all of the cells. Thus, dynamic flexibility can be reliably measured. Taking this example a step further, instead of thinking of these two trajectories as representing two different dyads, consider them as the same dyad in two different contexts. The dyad could have started with the flexible behavior depicted on the left of Fig. 2.5, but following a perturbation shifted their behavior to that depicted on the right of Fig. 2.5. This would be an example of reactive flexibility indicating an adaptation to environmental demands.

With this in mind, we explored both dynamic and reactive flexibility with mother–daughter dyads (Hollenstein and Lewis 2006). For this study, we used state space grids to measure the real-time dynamic flexibility of mother–daughter affect in an

A–B–A design of three discussion contexts: positive, conflict, positive. As expected, there was more negative affect during conflict and dynamic flexibility changed with the context—the dyads became more rigid during the conflict but returned to higher levels of flexibility in the final positive discussion. Thus, consistent with the effect of emotions on cognitive flexibility (e.g., Isen 2000; Matthews and McLeod 1985), we showed that the presence of negative affect actually restricted the dynamic structure of how that affect was expressed.

Variability, Flexibility, and Developmental Psychopathology

Interpreting variability as flexibility is also consistent with clinical observations that many psychopathologies are characterized by overly rigid behavior. My colleagues and I have explored this in several ways at several ages. In a large kindergarten sample, Hollenstein et al. (2004) showed that dynamic rigidity of parent–child affect during 2 hours of interaction was associated with concurrent levels of child internalizing and externalizing problems as well as the growth in externalizing problems over the course of 2 years. Moreover, partial correlations demonstrated that these differences in structural dynamics (rigidity) were not related to differences in the affective content.

Granic et al. (2007) compared externalizing children who had improved with treatment (self-regulation training and parent management training) to those whose externalizing behavior did not drop below clinical levels by the end of treatment. While there were no differences between these groups pretreatment, the parent–child dyads with improvers became significantly more dynamically flexible by the end of the treatment period This study also employed some innovations in region analyses based on four areas of the parent–child state space grid hypothesized to be the most important (see Fig. 2.6). These regions were mutual positivity, mutual hostility, permissive parenting, and mother attack—similar to the regions identified in the example for Fig. 2.2. There were no differences between improvers and nonimprovers in terms of duration in mutual hostile, permissive, or mother-attack regions, but improvers increased in mutual positivity from pretreatment to post-treatment. Moreover, using the same A–B–A discussion design as Hollenstein and Lewis (2006) described above, there was diminished reactive flexibility for the nonimprovers. These dyads could not repair from the conflict topic and shift to a positive topic without continuing to express negative affect (area *outside* of mutual positive region in Fig. 2.6). In the latest extension of this work, Granic et al. (2012) have shown strong correlations between *dyadic* flexibility and the amplitude of the child's electroencephalography (EEG) event-related potential called the inhibitory N2, even after controlling for many factors. The N2 is thought to be an index of inhibitory control, a key component in emotion regulation. Thus, the ability to inhibit behavior and emotions may be a key mechanism of affective flexibility.

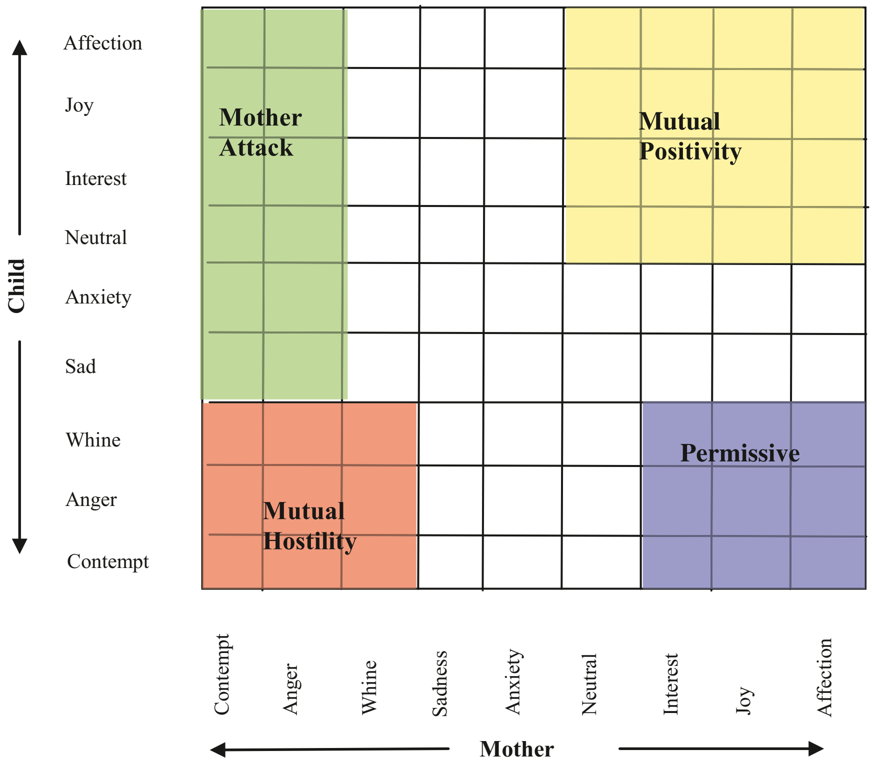

Fig. 2.6 Regions on a parent–child affect state space grid

Another study examining the link between flexibility and the development of psychopathology explored the interaction between positive affect and flexibility. Lunkenheimer et al. (2011) tested for the differences in positive versus negative flexibility in a sample of young children during interactions with mothers and fathers. During father–child interactions, flexibility and the flexibility by positive affect interaction predicted lower levels of externalizing behavior 2 years later when the children were 5 years old. Interactions with mothers revealed a different pattern. While the flexibility-positivity interaction predicted lower externalizing similar to the fathers' results, the direct effect of mother–child flexibility predicted higher levels of externalizing behavior. It is not yet clear why these results were different than in other studies. This was the youngest sample that has been tested for the flexibility-psychopathology link and these data were coded in a different way than most of the other studies reported here. Still, it begs the possibility of an upside-down u-shaped function of flexibility—too much or too little may be detrimental. What is needed is an examination of flexibility across more diverse contexts and pathologies.

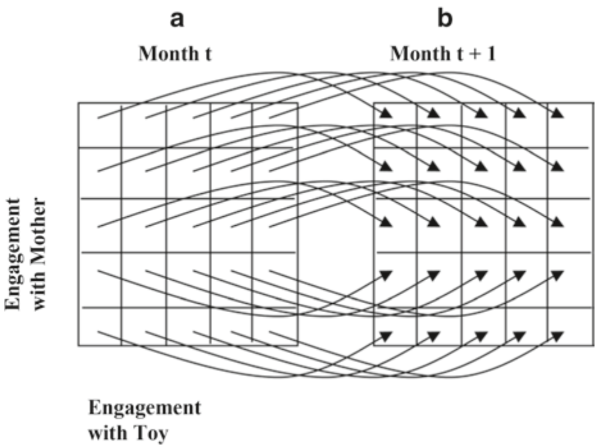

Fig. 2.7 Illustration of the calculation of the intergrid distance score (IDS). (Lewis et al. 2004)

Developmental Phase Transitions

Longitudinal studies testing the developmental phase transition hypothesis comprise the second set of studies on variability (Granic et al. 2003; Hollenstein 2007; Lewis et al. 2004). Here the general hypothesis is predicated on a stage approach to development that has long noted key transition periods. The first to be examined was the 18–20 month transition point, at which toddlers make a tremendous cognitive shift as they enter into the "terrible twos." This period is replete with a demonstrative "no!" dominating toddler communication as well as disturbances to sleeping and eating habits among other things. Lewis et al. (2004) observed toddlers once a month from 14 to 24 months of age as they engaged in two frustrating toy tasks. State space grids were created from one dimension of child's engagement with the toy (1–5) and engagement with his or her mother (1–5) who sat nearby reading magazines. Thus, each child had 11 trajectories, one for each month. The phase transition was hypothesized to result in greater month-to-month change during the 18–20 month window than either before or after. Monthly change was measured in two ways. First, we created an intergrid distance score (IDS) that was a calculation of the Euclidian differences between corresponding cells in adjacent monthly grids through the formula:

$$\mathrm{IDS} = \sum_{i=1}^{\#\text{ of cells}} (A_i - B_i)^2$$

where A_i is a cell from the state space grid from month t and B_i is a cell from the state space grid from month $t+1$, as shown in Fig. 2.7.

The second method for detecting month-to-month change was through a longitudinal cluster analysis. The 25 duration values of every grid for every month for every child were put into a single *k*-means cluster analysis to obtain a cluster membership for each grid. Grids that had similar patterns were grouped into the same

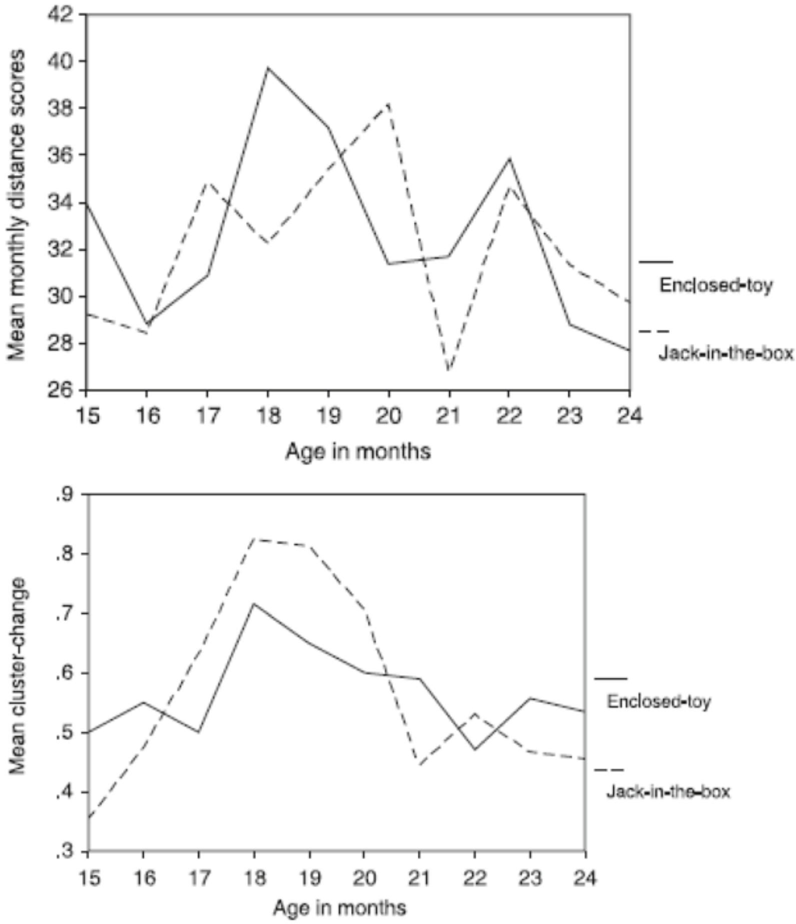

Fig. 2.8 IDS and cluster change scores from Lewis et al. (2004)

cluster. Then, for each child, a month-to-month cluster change score was created, giving a value of zero if the child stayed in the same cluster from time t to time $t+1$ and 1 if the child switched clusters from the previous month.

Using profile analysis, both the IDS and cluster change scores showed a quadratic pattern across months with the peaks occurring in the 18–20 month period (Fig. 2.8). Moreover this occurred with both tasks and despite their repetition over 11 months. This was the first study to provide dramatic evidence of a developmental phase transition using state space grids and it inspired two subsequent studies to test the phase transition hypothesis in adolescence.

Adolescence is one of the most tumultuous transitions in the lifespan, perhaps only second to the 18–20 month transition. The range and depth of changes in the physical, cognitive, social, and emotional domains certainly make this age period a leading candidate for a developmental phase transition. With a five-wave longitudinal sample of boys' interactions with their parents, starting at the age of 9–10 and continuing every 2 years, Granic et al. (2003) tested the adolescent phase transition hypothesis with state space grids. Two measures, the number of cells and the number of transitions, were derived from the 4×4 parent–boy grids. As expected, both measures peaked in the third wave when the adolescents were aged 13–14 years, the start of puberty and early adolescence for boys. Importantly this peak in variability was not just due to increases in conflict, as negativity peaked later, when the boys were 15–16 years old and variability had already dropped back down to pretransition levels.

To follow up this finding with boys, another study was conducted with girls to see if they went through the same phase transition 2 years earlier (Hollenstein 2007). Mother–daughter dyadic affect during conflict discussions were recorded over four longitudinal waves starting when the girls were in the spring of their grade-6 year (age 11.5 years) and every 6 months thereafter. All girls transitioned to a new school between the first and second waves. Possibly, because of the narrower time frame or the school transition, the results for the girls were not as straightforward as with the boys' study. Instead, only dyads with girls who did not experience stressful events concurrently with the school transition in wave 2 showed the hypothesized quadratic pattern. Interestingly, dyads with girls who experienced two or more stressful events concurrent with the school transition showed the opposite pattern. The quadratic was u-shaped because these dyads became more rigid over the transition period. It is possible that these girls (and their mothers) were resisting the change, clinging on to old habits in order to maintain emotional stability during a tumultuous period. Unfortunately, this study raised as many questions as it answered. Certainly more tests of developmental phase transitions are warranted.

Group Dynamics

Just as state space grid studies of individuals led to the examination of dyadic systems, there is an interest among many developmental scholars to tap the group dynamics of peer interactions. Martin et al. (2005) created some solutions to this problem in a study that recorded the gender and social competency category (competent, externalizing, and internalizing) of the target child's play partners, as well as the target child's emotional valence, on the playground across an entire preschool year. Of course, pragmatic issues led to the observations being random and distributed throughout the year. Hence, transitional measures were not realistic, but measures of states were. The main finding was in support of increasing homophily across the year: girls increased their play with girls and boys with boys across the year. There were also interesting distributions of play partners by competency. This study provides one approach to the analysis of *n*-dimensional systems. Another approach was taken for the analysis of coach–athlete interactions with a team of athletes (Erickson et al. 2011). In this study, the coach and each athlete was coded continuously during

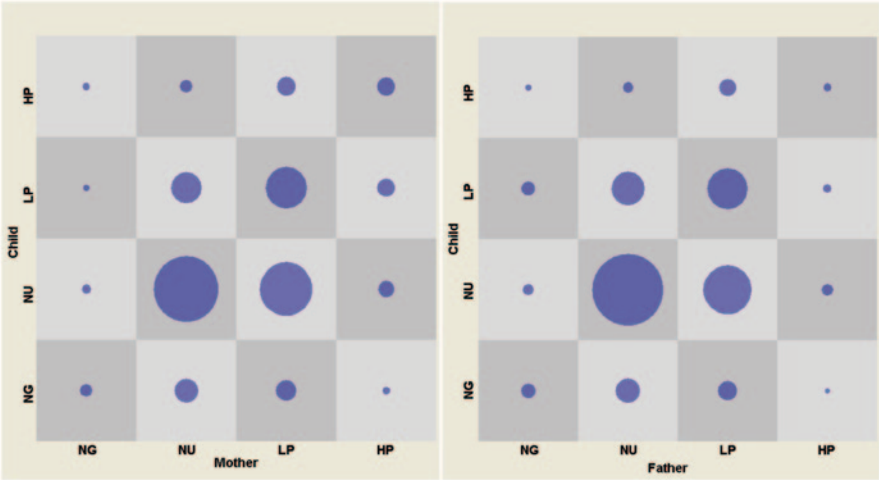

Fig. 2.9 Summary state space grids of proportional durations in each cell for mother–child (*left*) and father–child (*right*) interactions. (Lunkenheimer et al. 2011)

practice sessions, with a state space grid dimension corresponding to each. From this starting point, each coach–athlete combination could be analyzed and the athletes could be combined to go beyond two dimensions. Further discussion of ways to go beyond two dimensions is covered in Chap. 7.

Summary Grids

State space grids have also been used to depict static visualizations of summary information. Similar to a 2×2 matrix of values, these grids use a single node, centered in each cell, to depict the magnitude (e.g., mean duration or frequency) of the activity in that cell for subgroups or for an entire sample. In the Lunkenheimer et al. (2011) study, the proportional durations in each parent–child affective state were displayed for mother–child and father–child dyads (Fig. 2.9). Comparing these two grids, it is easy to quickly get a sense of how similar and how different these two groups were. Although it was not presented in the original Martin et al. (2005) study, Fig. 2.10 shows a summary of the play–partner combinations across the year (Hollenstein 2007). Here, the size of the plot point indicates the relative frequency of that particular target child with peer competency-type combination. With this display we can see that competent girls tended to play with each other and externalizing boys tended to play with everyone. The means to create these summary grids will be described in Chap. 5.

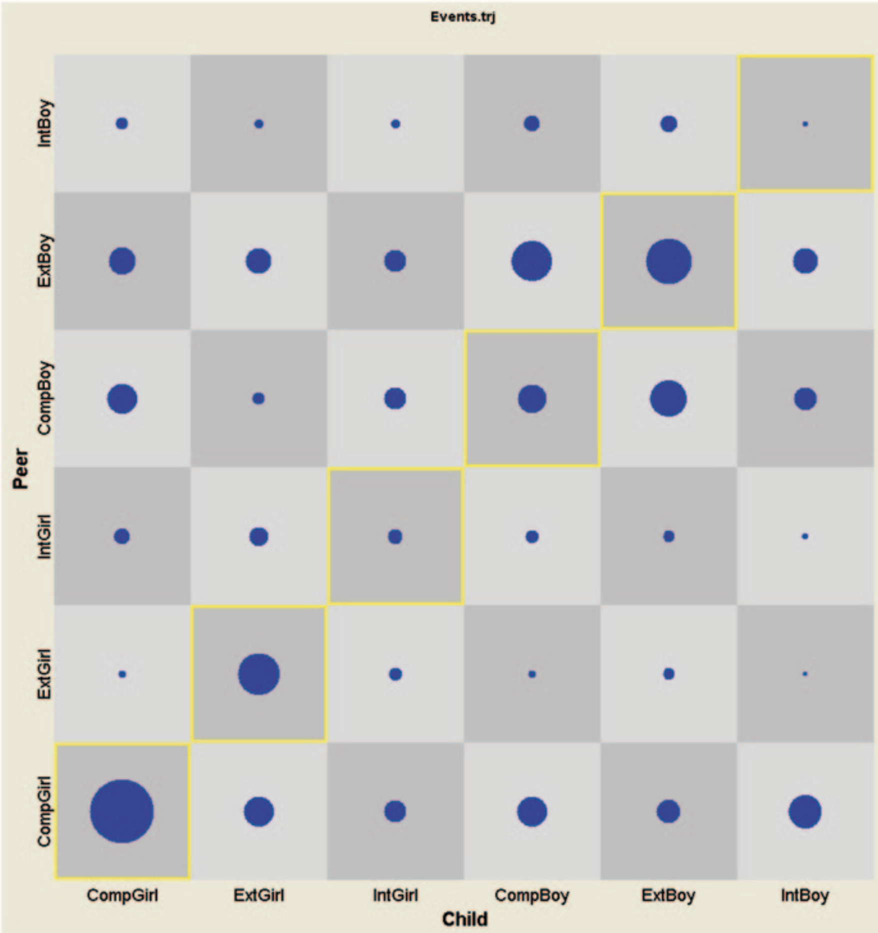

Fig. 2.10 Summary of preschool playground peer interactions from the Martin et al. (2005) study (from Hollenstein 2007). The target child's identity (gender and competency category) is on the *x*-dimension and the identity of the peer play partner is on the *y*-dimension. The size of the plot point indicates the relative frequency of that play combination across the year. The *highlighted diagonal* indicates homophily

State Space Grid Studies: Unpublished or in Preparation

The studies published till date represent a fraction of the state space grid analyses that have been conducted till date. Some of these are in the process of reviewed or prepared for submission, some were just used for conference presentations, some are languishing in file drawers somewhere, and others were used in studies that ultimately did not report using grids for simplicity. Through consultations, workshops, and collaborations, I have worked on no less than 100 data sets to create state space

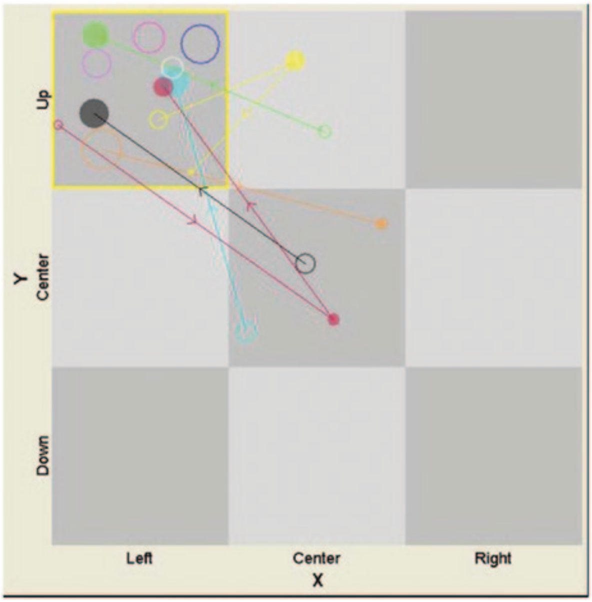

Fig. 2.11. Example of an individual eye gaze state space grid. Trajectories indicate the direction of gaze following each of the ten questions that required episodic recall for one subject

grids. Through these experiences, I have gathered a range of different grid types and techniques that depart from the dyadic emotion grids that have been my personal mainstay. Seeing the diversity of approaches is always one of the highlights of a workshop—I can see the light bulbs turning on. I hope these descriptions in the following sections will do the same here.

Eye Gaze The first variation to explore is using state space grids for tracking eye gaze trajectories. Individual eye gaze grids plot the direction of gaze using the x-dimension as left–center–right positions and the y-dimensions as down–center–up positions relative to a focal target. McCarthy et al. (2007) documented the direction of gaze at an experimenter who asked questions that required episodic recall (see Fig. 2.11). Most people in western cultures look up and to the left while accessing a memory prior to responding. This provides a social cue to others to not interrupt while thinking. McCarthy et al. showed that this eye gaze tendency does not reliably occur before the age of 7.

In a dyadic eye gaze study, van Straaten et al. (2009) examined the eye gaze patterns of males and females having a discussion with an opposite sex confederate who was independently rated as either attractive or average-looking. Although state space grids were used for analyses, these were not reported in the final journal article. This was a 2×2 grid of a binary categorization of gaze: looking directly at the other person's face or away from their face (see Fig. 2.12). Compared with average discussion partners, males tended to have longer staring events at the face of attractive confederates. In contrast, females tended to have more frequent but extremely short mutual gaze durations with attractive partners compared with average

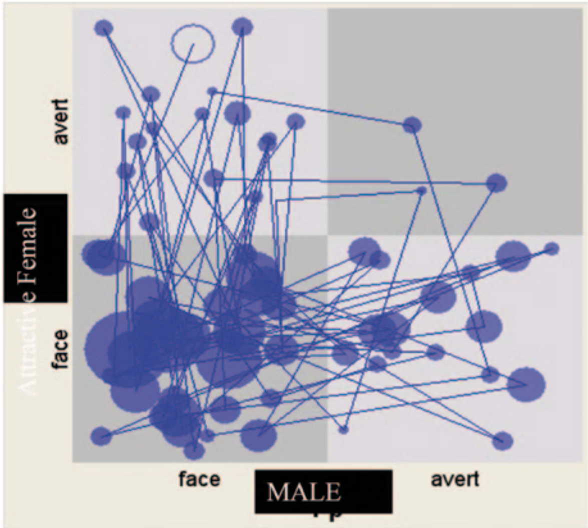

Fig. 2.12 Dyadic eye gaze state space grid of a male with an attractive female confederate

partners. While this study demonstrated clear gender differences in interaction with attractive strangers, it also demonstrates for our purposes here that the minimum-sized 2×2 state space grid can provide insightful information about behavioral dynamics.

Diaries and Experience Sampling Another increasingly common research method is the examination of emotions or behavior over the course of days, weeks, or months. These studies may use diaries or electronic devices through what has been called the experience sampling method (ESM; e.g., Larsen and Csikszentmihalyi 1983) or ecological momentary assessment (EMA; Silk et al. 2011). The big difference here is that, rather than time units of seconds or events that occur within a short window of time, the intervals between events could be 24 hours or more. Trajectories in these instances would reflect sequences of daily assessments and would be useful for tracking such things as mood or developmental progressions that occur over the course of weeks, months, or years.

In a creative study with a nested longitudinal design, Lichtwarck-Aschoff et al. (2007) obtained daily diary reports from mothers and adolescent daughters over the course of 2 weeks, repeated every 6 weeks for 18 months. Although the dyadic results are unpublished as yet, the grids used for exploratory analyses are worth considering. In Fig. 2.13 there are three different dyads who report how they felt during the discussion and then how they felt later before going to bed. Thus, each trajectory is a two-step sequence indicating the degree to which both the mother and the daughter repaired (moved from negative to positive emotional states) from conflict. Vertical trajectories indicate instances when the mother but not the daughter repaired, horizontal trajectories indicate instances when the daughter but not the mother repaired, and diagonal trajectories indicate instances when both the mother

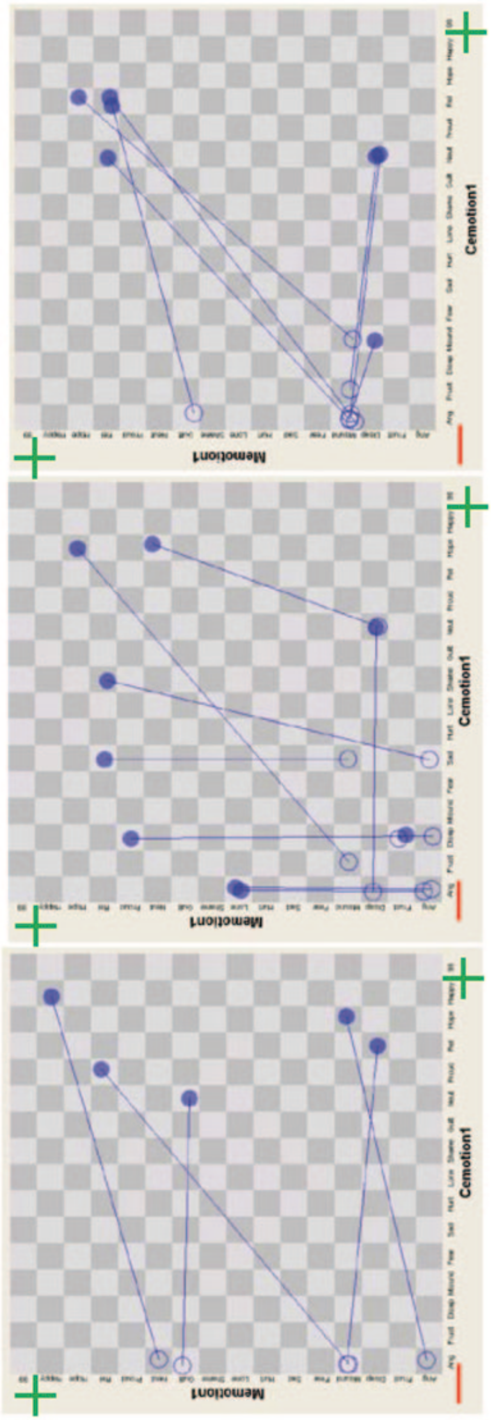

Fig. 2.13 Three example state space grids from a mother–daughter diary study. The *x*-dimension corresponds to the daughter's report and the *y*-dimsension corresponds to the mother's report. Plot nodes indicate the most intense emotion during a conflict (*open nodes*) and then after the conflict before going to bed (*closed nodes*). Emotions are arranged on the dimensions in a quasiordinal fashion according to valence, with the most negative emotions to the *bottom and left* and the most positive to the *top and right*

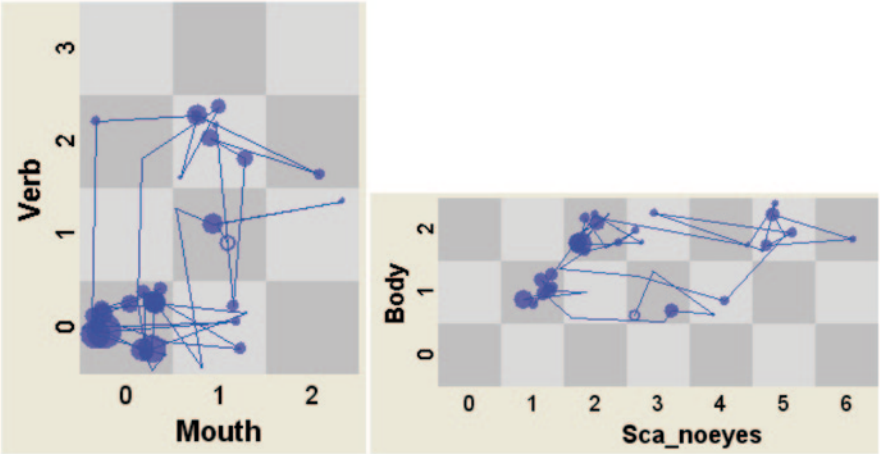

Fig. 2.14 Using state space grids to compare code categories. The state space grid on the *left* shows the relationship between the mouth tension and verbal tension codes for one participant, the one of the *right* shows the relation between the body tension code and the overall self-conscious affect score made from the combination of all the ordinal state variables (except for the Eye Gaze category)

and the daughter repaired. This example illustrates how even short trajectories can be used in state space grid analyses as well as how data at two time scales (within a day and across days) can be analyzed simultaneously with state space grids.

Measurement Analysis As many applications of state space grids involve coded observational data, it may be useful to examine the relations between code categories as part of the development of the measure. My students and I have developed a self-conscious affect coding system for participants giving a speech in the laboratory, for example, for which there are four primary categories, each of which vary in intensity (Lanteigne et al. 2011). That is, each code category is a state variable with ordinal levels of intensity as the states. Thus, we have a complete data stream for four domains of self-conscious behavior: Body Tension (none, medium, and high), Mouth Tension (none, medium, and high), Eye Gaze (direct and averted), and Verbal Tension (none, medium, silent, and high). The goal is to create an overall self-conscious affect score from the combination of these state variables and to compare these to self-reported self-conscious affect and psychophysiological measures. As part of this research, we examined the relations of the individual code variables to each other and in relation to other measures. Figure 2.14 shows a couple of state space grids derived for this purpose. Here, we can see how the dynamic states relate to one another in real time and how they contribute to the overall code score. From these explorations, we have since developed a revised version of the coding system.

We also were interested in the dynamic relations between self-conscious behavior and each person's appraisals of their own distress. Following the video-taped

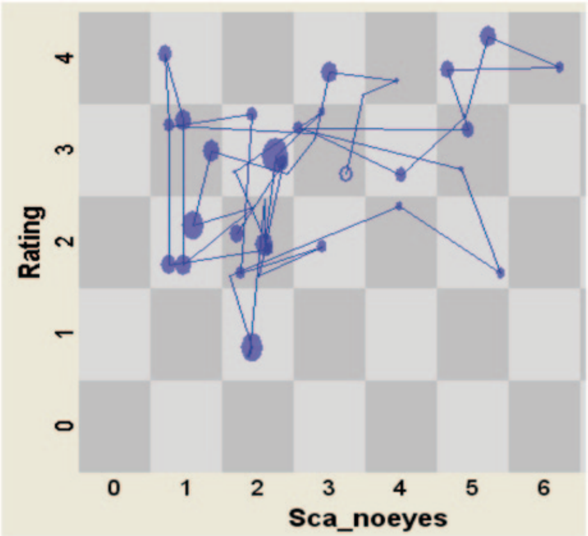

Fig. 2.15 Using state space grids to compare code categories to dynamic self-report. The participant's real-time rating of their own distress is synchronized with their observed self-conscious affect

speech, participants watched their own speech on video and rated it along a single dimension of distress intensity. As shown in Fig. 2.15, we can then look at the psychological and behavioral dynamics simultaneously. The example shown here is of someone who was fairly distressed and moderately concordant across the psychological and behavioral domains.

Psychophysiology With the growing ease of measuring and analyzing psychophysiological processes in vivo, more and more researchers have detailed time series data that are ideal for state space analyses. The primary objective is to relate psychophysiological measures such as heart rate, galvanic skin response, and electrodermal activity, to behavior and psychological appraisals in real time. This is no simple task. State space grids can be adapted to perform some analyses that can be useful, though it requires turning continuous measurement into categorical values. As Chap. 7 covers this topic, I would not elaborate too much here. Instead, briefly, Fig. 2.16 shows an example of how the combination of self-report and physiological measures can be used for deeper understanding of coordinated behavioral processes for two participants who gave speeches as described in the previous section. In these examples, we see evidence of discordance between physiological and psychological measures. This discordance in emotional situations can reveal differences in emotional processing related to emotion regulation and the development of psychopathology (Lanteigne et al. 2012).

Triads Though state space grid analysis through GridWare (described in the subsequent chapters) is based on two-dimensional space, there is a need to go beyond two state variables for many research questions (e.g., Martin et al. 2005). The shift from two to three dimensions is tricky for many scientific endeavors (Diacu 1992), but

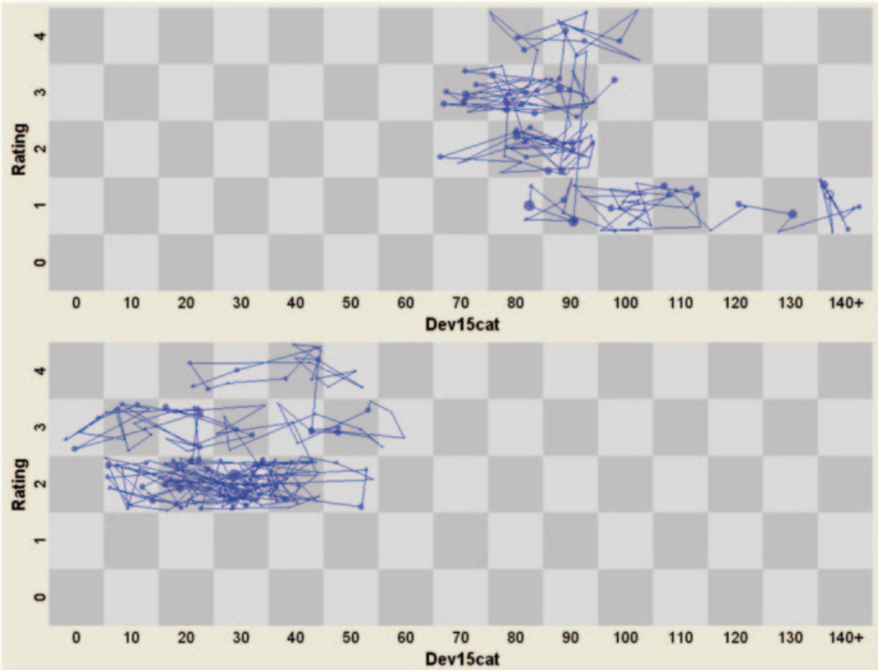

Fig. 2.16 State space grid of self-reported distress and percent change in heart rate. The state space grid on the *top* shows a high distress with low arousal participant and the one on the *bottom* shows a low distress with high arousal participant. Dev15Cat = 15 categories of percent deviations from baseline heart rate

with a few modifications state space grid analysis can accommodate other dimensions beyond two. From the interactions among three people (e.g., mother, father, child) to the coordinated dynamics of multiple measures (e.g., heart rate, self-report, and behavior), there are many instances for which a three-dimensional state space is desired. This is the focus of Chap. 7, so I will forego repeating the details here. It is sufficient at this point to note that state space grid analysis is flexible enough to accommodate many types of variations in state variables.

Lag Sequential Analysis Finally, although much of this book is dedicated to the analysis of simultaneous events, state space grids can also be used to analyze lagged events. In some cases, for example, researchers want to know the effect of one person's behavior on another (e.g., mother and child). Thus, it is important to quantify these lagged events. With state space grids, this can be done by plotting the trajectory with one dimension representing the original sequence and the other dimension representing the same sequence starting one event later. Lagged state space analysis is covered in more detail in Chaps. 5 and 7.

Conclusion

The contents of this chapter should give an indication of the depth and breadth of the state space grid technique. To date, we really have only scratched the surface. On one level, the technique is a deceptively simple variation of plotting data by embedding time within the two-dimensional space. However, just that small twist opens up worlds of possibilities. With the completion of these first two chapters, you now have all the background necessary to be able to start using state space grids yourself. In the next two chapters, I will describe the state space grid software, GridWare. First, I will walk through the software with a simple example and in the subsequent chapter I will describe how to set up your own projects.

Chapter 3
GridWare

In this chapter, I describe the software for making state space grids, GridWare (Lamey et al. 2004). GridWare is a versatile visualization and data manipulation tool for multivariate time series of sequential (ordinal or categorical) data. It was developed as an exploratory data analysis tool as well as a source for a range of measures derived from the state space grid formatting. Since 2004, I have managed this software and its continued development, including the website for its free distribution: www.statespacegrids.org. You should go to that website now and download the version appropriate for your platform. I will be using examples of state space grid analyses throughout the remaining chapters, so it will be useful to be able to work along with the sample data provided in the download. Note, the program runs in Java and it may be necessary for you to download Java or upgrade to the newest version. It is fast and simple from www.Java.com. Do not worry. Java is perfectly safe and runs many of the web pages you look at on the internet. These days, it is almost always loaded on new computers by default (Fig. 3.1, Box 3.1).

Box 3.1: GridWare: Free and Open Source GridWare is a free and open-source program. It was first codeveloped by a brilliant programmer, Alex Lamey, who shared my view that research output and research tools should remain in the public domain. All of the funds that have been used to develop GridWare have come from the public sector—allocations from government tax revenues distributed as research grants, fellowships, and university funding. Thus, it is important to make sure there are no barriers to the public's access to the fruits of research, whether it is a journal article, book, or software program. In some ways, this has slowed the development of the software, but it has maintained the integrity of knowledge in the public domain.

Open-source software is quite common and was developed out of a larger view that the internet and computer applications should be public domain. This has lead to great advances in software as many experts can contribute bit by bit to making better software tools. Users interested in the source code for GridWare can find it on Source Forge (http://gridware.sourceforge.net/).

T. Hollenstein, *State Space Grids,* DOI 10.1007/978-1-4614-5007-8_3,

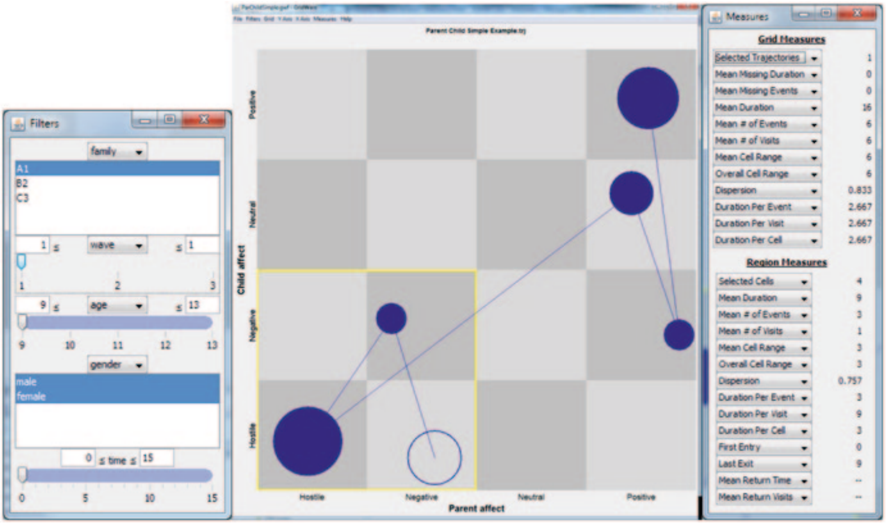

Fig. 3.1 GridWare windows

> GridWare has a "copyleft" via the Gnu Public license. For more information about open source on the internet, see http://opensource.org.

When you download GridWare, just follow the instructions provided on the website. You can save the program and the files that come with it anywhere on your computer, but I recommend saving it with the rest of your programs or high up in your hard drive's hierarchy (e.g., C:\\GridWare\) to make it easy to access as you work. You will be able to save your GridWare projects anywhere on your hard drive.

Getting Started with GridWare

The way that GridWare works is through two types of files that the user sets up. The first file type is a single control file, the GridWare File, which tells the software about the dimensions of the state space, the categories in those dimensions, and information about the files that are included in the project. The other file type is a Trajectory File and each project typically has many Trajectory Files. Figure 3.2 illustrates the general set-up. Details about each of these files will be provided in a later section.

After all the files have been extracted to your hard drive, open GridWare by double-clicking on the program. The name of the program will depend on the version and your operating system (e.g., GridWare 1.1b.exe or GridWare 1.15a.jar). Next select the GridWare File (e.g., Example 1.gwf) that corresponds to your project and double-click or click the "Open" button. Depending on the size of the project

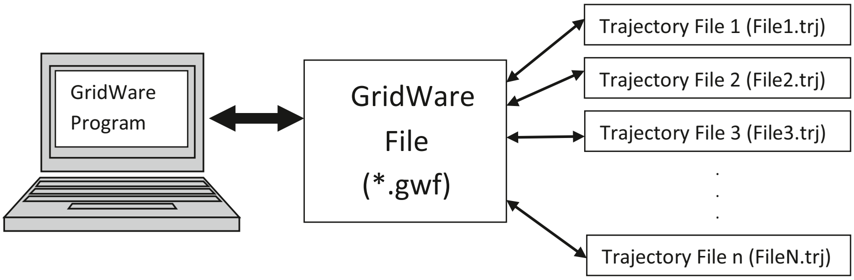

Fig. 3.2 GridWare file structure

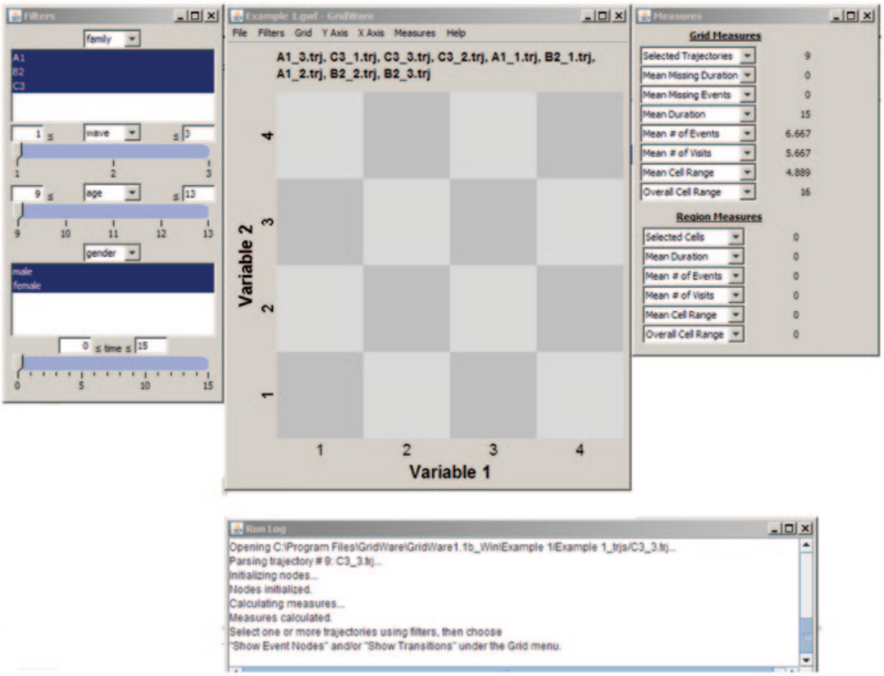

Fig. 3.3 Example 1 upon first opening

it may take a few moments for the project to open. Before describing GridWare any further, it will be best to open one of the example projects that comes with the GridWare download now.

Open “EXAMPLE 1.GWF” in GridWare now

Once open, Example 1.gwf should look similar to Fig. 3.3. In order to save computer memory, the trajectories are not displayed upon opening by default. GridWare has four windows. The largest is the Grid Window, which has the state space grid that displays the trajectories as well as the menu bar. The Data Filters Window allows the user to select which trajectories to display based on the trajectory variables. The Measures Window displays the measures for the whole grid and individual cells, if selected. The fourth window is the Run Log Window that displays the messages related to the processes and functions performed in the running of GridWare as well as any error messages to aid in troubleshooting. Using the Example 1 project you have open, let us walk through some of the key features. Greater detail can be found in the GridWare User Manual that comes with the downloaded software (Box 3.2).

Box 3.2: Glossary of GridWare Terms

Duration-based Data	Trajectories for which information about the time of onset for each event is recorded in the Onset Variable. An Event's duration is the difference between an Event's onset and the onset of the next Event.
Event	A row of data in a Trajectory File. Events are the smallest unit of states displayed on a grid.
Event-based Data	Trajectories for which each onset value after the first Event does not represent the offset of the previous Event. Most often, the Onset Variable values are a sequence of integers representing a count of the number of Events.
Filter Variable	A "between-trajectories" variable that identifies or differentiates Trajectory Files. Typical Filter Variables include participant ID, sex, age, experimental condition, longitudinal wave, or task.
GridWare File	The control file that provides all the information GridWare needs to create the state space grid and import the Trajectory Files. It is always named with a .gwf file extension.
GridWare Project	The combination of a GridWare File (e.g., project1.gwf) and a set of Trajectory Files saved in a folder of the same name as the GridWare File plus "_trjs" (e.g., Project1_trjs) in the same directory.
Final Offset	The final value of the Onset Variable in every Trajectory File that indicates the end of the final state. There can be no text to the right of or below the Final Offset value. *See Onset Variable.*
Onset Variable	The first column of every Trajectory File that reflects the time value of the onset of each event. For event-based data, onset values are simply integer counts of the sequence of events. For duration-based data, each onset after the first also functions as the offset of the previous event. *See Offset*.

State Variable	A single dimension of the state space grid that has two or more categories (*see State*) and corresponds to a column of data in the Trajectory Files.
State	A category of a State Variable.
Tab-delimited Text	A text file in which columns are indicated by a tab as a separator. It is one of the most common formats of text files, especially for data, that can be imported into virtually every spreadsheet or data analysis software program. All input and output files in GridWare are tab-delimited.
Trajectory	A sequence of Events comprised of the joint States of two State Variables that can be depicted on a state space grid. In terms of a Trajectory File, it is the sequence of Events in two columns.
Trajectory File	A Tab-delimited Text file that contains, at minimum, a first column labeled "Onset" that contains the onset values for each Event, and two other columns of State Variables. A Trajectory File can have more than two State Variables. All Trajectory Files are named with the .trj file extension.
Trajectories List	Tab-delimited Text matrix or file of one row per Trajectory File and, at minimum, a column for an identifying variable (e.g., participant #) and a second column that must be named "Filename" for the names of each Trajectory File in the project. All variable columns other than Filename are Trajectory or Filter Variables.
Trajectory Variable	*see Filter Variable*.

Exploratory Data Analysis with GridWare

The Grid window is the main screen for GridWare. It displays trajectories on a two-variable state space. In this case, we start with Variable 1 on the *x*-axis and Variable 2 on the *y*-axis. Each state variable has four states or categories: 1, 2, 3, and 4. This is a nondescript example, so these states could correspond to a wide range of possible variables. For instance, these could correspond to ordinal levels of distress (1 = low, 4 = high) or nominal categories of behavioral states (hostile, withdrawn, neutral, and positive) of the same individual or of different individuals (mother and child). It may be helpful for you to imagine relevant variables in your own research domain that could be represented by these State Variables as you work through this example. You could even do that right now by holding down the Alt key while clicking on one of the numbers on the *x*-axis. In axis-label edit mode, you can change these labels to whatever you want. Once done, hit return to leave axis-label edit mode. This is not the best way to make these changes permanent, but it may help you as you work through the example for now.

The most important first thing to do with Example 1 is to display the data. This is a small project (nine short trajectories), so it is not a problem to do so. With larger projects, you will want to select one or only a few trajectories first before displaying. This is not only to save the time it would take to load all of them at once, but also because after a certain number of trajectories are plotted on the grid at once it can look a bit messy. On the Grid menu in the Grid Window, there are four sections pertaining to the layout type, display features, nodes, and transitions. Select Show Event Nodes to display the nodes or plot points and then go back to the Grid menu to select Show Transitions. Now you can see all the nine trajectories simultaneously. Note that the diameter of the node corresponds to the duration of the event. Also note that within each cell, the position of the node is random—each time you open or select a new trajectory the software uses a random number generator to determine the x/y coordinates for each event within each cell. For purely display purposes, you can rerandomize these positions by selecting Randomize Nodes from the Grid menu.

The next thing to do is to select one trajectory from the Filters Window to display. There are five Filter Variables: Family, Wave, Age, Gender, and, by default, Time. Family and Gender are categorical Filter Variables and so are displayed as a list of values in a scrollable box (though with so few elements in this example, no scrolling is necessary). Even with ordinal filters, I often declare them as categorical because it provides greater flexibility in selecting nonadjacent values (just as with most computer lists of elements, you can use Ctrl or Shift keys to select specific elements or a range of elements, respectively). In this example, Wave and Age are ordinal Filter Variables and therefore have sliders that you can manipulate with the mouse or a combination of Ctrl, Alt, or Shift keys with the mouse click. For example, hold down the Alt key while clicking on the slider button for the Wave variable and slide it to the right. Now only Wave 3 trajectories are displayed on the Grid Window. Clicking on the slider line to the left of the slider button now moves it to the left, clicking to the right moves it to the right.

Now that you know how to select trajectories, select just Family A1 at Wave 1 (as shown in Fig. 3.4). Select Randomize Nodes from the Grid menu a few times until all of the nodes are displayed in a way where you can see each distinctly. By default, GridWare displays the first node in the trajectory sequence as an open node (no color filled in), though this can be changed in the Grid menu. With this trajectory, there are nine events, each corresponding to a node. Because there is a third State Variable not shown (we will get to this later), we know that there are repeating events—two or more nodes, sequentially connected by a transition line, in the same cell. Briefly, this is a circumstance in which the GridWare measure of Events is not equivalent to Visits (a transition from one cell to another). In this one trajectory, for

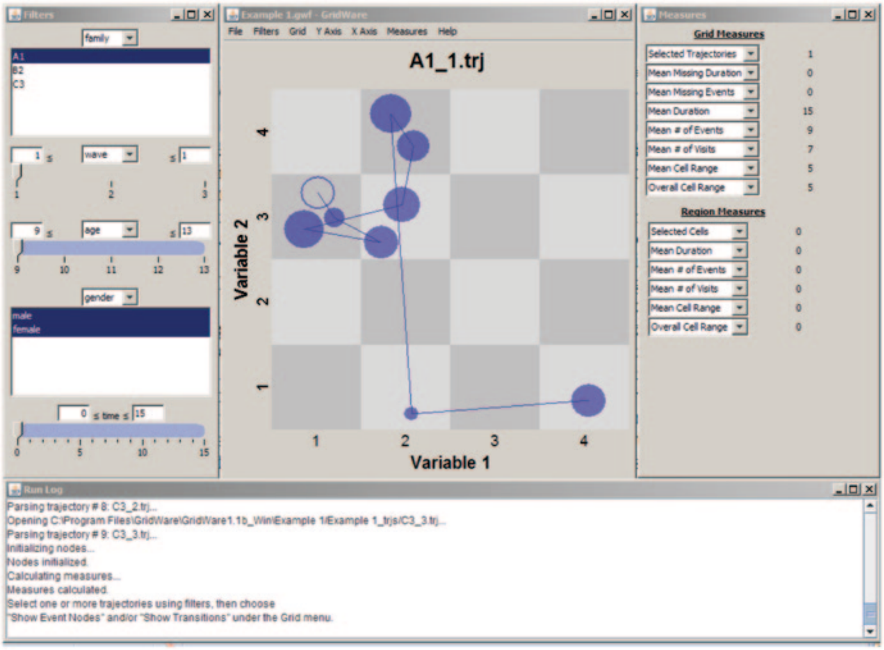

Fig. 3.4 The single trajectory of family A1 at Wave 1

example, we have nine Events but only seven Visits. This Event/Visit distinction is important and something we will return to later.

Sometimes, especially with long trajectories with many events, it is preferable to know whether an event occurred in the middle or toward the end of the sequence. For this reason, it is possible to change the display to a diagonal layout (option on Grid menu). With trajectories displayed this way, the sequence of events are plotted along a diagonal line from bottom left to top right within whatever cell the event occurred. The nodes in the bottom left portion of the cell in which they are plotted are those from the early parts of the trajectory sequence, while those in the top right portion of the cell in which they are plotted are those from the last of the trajectory sequence. This is only a display variation and has no effect on the measures derived. To continue with the walk through of GridWare and Example 1, return the display to the Random Layout setting as shown in Fig. 3.4.

Data analyzed through GridWare is dynamic, it changes over time. That is why the program includes a Time slider by default. While only trajectory A1_1 is displayed, click in the right Time box that displays the number 15. That number corresponds to the maximum length of any one of the trajectories in the project—these are simulated data corresponding to 15 s of event sequences. When you click in that box, notice how the slider button switches sides. This is how you can control which end of the slider you would like to move. Now that the button is on the right, hold down the Alt key while you click and slide it back and forth, left and right (you can also use the arrow keys at this point, if you like). This is how you can explore the trajectory as it unfolds over time. This is one of my favorite features and can be used in combination with screen capture software to make a movie of the trajectory. If you do not want to see the cumulative trajectory, you can also just highlight a time window and move that along. To do that, move the slider so that only 0–3 s are displayed (you can also do this by typing in those values in the boxes and hitting return). Then, if you hold down the Ctrl key while moving the slider to the right, it will retain that 3-s window but move it along one time unit at a time. If you hold down the Shift key, it will progress through adjacent time windows, 0–3, 4–7, etc. Time sliding is not as dramatic with these short trajectories but with trajectories corresponding to longer durations (e.g., 10 min parent–child interaction), it can be quite insightful.

If you are like me, you have already started playing around with Example 1 and selected various combinations of Filter Variables to explore and compare the nine trajectories in this project. This exploratory capability is one of the best features of GridWare and often results in one of the first insightful moments experienced by new users. For this reason, GridWare also includes a coloring option to facilitate visual comparison of trajectories. To show off this feature we first want to go back to displaying all nine of the trajectories at once. There are several ways to do this. The manual method would be to click on the "A1" in the Family Filter Variable box and then either hold down the Ctrl or Shift key while selecting the rest or simply hitting control-a (command-a on Mac) to select all of the elements in the Family box. Then, while holding down the Alt key, click and slide the Wave slider button so that all three waves are selected. Alternatively, you can use the automatic method by selecting the Reset Filters option from the Filters menu on the Grid Window. The reset option is quite useful after a bit of exploring as you may not recall which filters to change (especially if you have added or removed Filter Variables from the display by using the drop down menus in the Filters Window or the Add Filters option on the Filters menu in the Grid Window).

Once all the trajectories are selected, now select just the Wave 2 trajectories. In the Grid menu, select the "Colour Nodes" option. This brings up a window with 256 colors. Whatever color you click on will now be the color of the trajectories selected in the Filters Window. Click on one of the reddish colors at the center of that color matrix and OK. Now, switch the Filters Window selection to all of the Wave 3 trajectories and make all of them dark green. So, now we have color-coded all the trajectories such that Wave 1 is blue, Wave 2 is red, and Wave 3 is green. Go back to the Filters Window and select just Family A1 and also all of the waves. Now, we can see how Family A1 behaved differently at each wave. Try the same for families B2 and C3. You should see something similar to Fig. 3.5.

Through the control of the display options in GridWare, exploratory data analysis is limited only by the user's imagination. For example, color can be used as a way of analyzing beyond two dimensions (see Chap. 7) or, in combination with Time slider, as a means of depicting time epochs. Before moving on to describe the measures derived from state space grids, it is important to note two final features. First, grid images can be exported as a .png file using the option in the File menu.

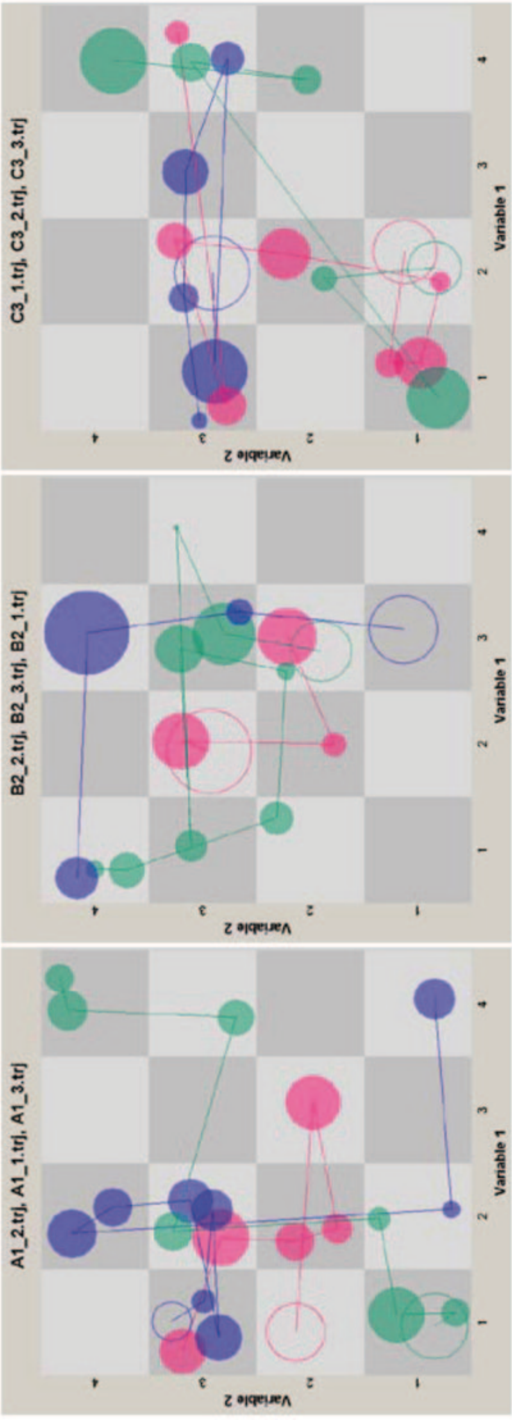

Fig. 3.5 Example 1 trajectories by family using color to denote Wave (*1* = blue; *2* = red, *3* = green)

Image quality is best if you first maximize the Grid Window to full screen and then select Square Cells from the Grid menu. Selecting opaque rather than translucent nodes via the Grid menu may also make the nodes stand out with more contrast. However, this may also make it difficult to distinguish individual nodes.

Finally, there is no need to save your GridWare project every time. Saving will create a new, duplicate project with the node position and color saved for each event in each Trajectory File. That is, each Trajectory File will have newly added tab-delimited columns for the x-coordinate, y-coordinate, and RGB color code. With these saved, the project will open with exactly the same look as when you saved it. I do not often save my projects but when I do it is for presentations or even for customized grids such as the summary grid in Fig. 5.4.

Measures

Beyond exploratory data analysis, the other purpose of GridWare is to provide measures of various aspects of the dynamics of the trajectories. These indices range from the very simple to the computationally complex. The Measures Window provides updated values of most of the available measures as you conduct exploratory data analyses through Filter Variable selection. However, the typical user wants to use these measures in statistical models in order to be able to make inferences about real-world populations (see Chaps. 5 through 7). Thus, GridWare offers a range of options for exporting these measures for use in any spreadsheet and statistical software package.

Measures are divided into two basic types: Whole-grid Measures which are derived from the information across all the cells, and Cell or Region Measures which only apply to individual cells or regions selected by the user. Thus, the values of Cell or Region Measures in the Measures Window are blank or missing if no cells are selected. The user can also select which measures to display in the Measures Window. In the Measures menu in the Grid Window, the user can select which type of measure to add: Whole-grid or Cell/Region. As with the Filters menu option, this action will insert a measure location in the Measures Window. By default, upon opening GridWare, only eight Whole-grid Measures and six Cell or Region Measures are displayed in the Measures Window. By using the Add Measures option, the full complement of 14 Whole-grid and 14 Cell or Region Measures can be displayed. There are two versions of GridWare with slight differences in the available measures (see Box 3.3).

Box 3.3: What Is the Difference Between Versions 1.1b and 1.15a? The original version of GridWare was 1.1b, the newer version is 1.15a. The newer version has everything that the old version had with one notable exception. The export measures dialogue box options are different. Most notably, it is not possible to export individual cell values. The new additions in version

1.15a are measures of entropy and transitional propensities. Mac users should also note that the older version does not work with OS 10.5 and later, but the newer version 1.15a works fine.

Whole-Grid Measures

Most of the Whole-grid Measures are indices of within-trajectory variability, though there are a few that provide basic descriptive information (see Box 3.4). In this latter group are the count of the number of trajectories currently displayed in the Grid Window, the total duration or trajectory length of the currently displayed trajectories, and the count and duration of missing events. The rest of the measures use some combination of duration and frequency information within each cell that is then averaged or summed across cells. If only one trajectory is displayed, these are the raw values for that one trajectory. If more than one trajectory is displayed, then the values are the means across all of those trajectories. One notable variation to this rule is that Overall Cell Range reflects the full range across the entire set of displayed trajectories. For example, if trajectory A visits only cells 1 and 2 but trajectory B visits cells 1, 2, 3, and 4, the Overall Cell Range would be 4 but the Mean Cell Range would be 3.

Box 3.4: Whole-Grid Measures

Selected Trajectories	The number of individual trajectories currently displayed.
Mean Missing Duration	The total duration of all the missing events within the trajectories displayed.
Mean Missing Events	The total number of missing events within the trajectories displayed.
Mean Duration	The average duration of all the trajectories currently displayed.
Mean # of Events	The average number of events of all the trajectories currently displayed. An event corresponds to a single line in a trajectory file, and to a single node in the onscreen display.
Mean # of Visits	The average number of visits to each of the cells across all the trajectories displayed. A cell visit is one or more consecutive events occurring within a single cell of the currently visible grid, beginning upon a trajectory's entry into the cell, and end-

	ing upon its exit. Thus, the number of transitions between cells = visits − 1.
Mean Cell Range	The average number of cells visited across all the trajectories.
Overall Cell Range	The total # of cells visited by at least one trajectory across all trajectories selected. This value is equal to Mean Cell Range for a single trajectory.
Dispersion	The sum of the squared proportional durations across all cells corrected for the number of cells and inverted so that values range from 0 (no dispersion at all–all behavior in one cell) to 1 (maximum dispersion) created by the formula: $$1 - \frac{\left(n \sum (d_i / D)^2\right) - 1}{n - 1}$$ Where D is the total duration, d_i is the duration in cell i, and n is the total number of cells.
Duration per Event	The mean, across trajectories, of the duration of each trajectory displayed divided by its number of events.
Duration per Visit	The mean, across trajectories, of the duration of each trajectory displayed divided by its number of visits.
Duration per Cell	The mean, across trajectories, of the duration of each trajectory displayed divided by its cell range.
Entropy Measures (version 1.15a only)	Two types of entropy measures are calculated using probabilities based on duration or cell visits. In both cases, if P is this probability, then entropy is calculated according to the Shannon and Weaver (1949) formula: $$\sum (P_i * \ln(1 - P_i))$$ where i is an index of each cell on the grid and P_i is the probability in cell i. For Duration Entropy, P_i is the total duration in cell i divided by the total duration of the entire trajectory. For Visit Entropy, P_i is the number of visits to cell i divided by the total number of visits in the entire trajectory.

Cell or Region Measures

Using the mouse to select one or multiple cells enables GridWare to calculate measures for only those selected cells. A region is simply two or more selected cells and though they are most often adjacent cells, they do not need to be. As described in Box 3.5, selecting a subset of the state space grid allows for an additional set of measures to be calculated beyond those in the Whole-grid Measures section. All of these additional measures are ways to quantify the attractor strength of that cell or region. First Entry and Last Exit are fairly straightforward and can be used in a variety of circumstances, but the Return variables are a bit more subtle.

Box 3.5: Cell, Region, and Transitional Measures Most of the Whole-grid measures can also be obtained for cells or regions (see Box 4.4). The following are only those measures that are uniquely suitable for cell or region analyses.

Mean Return Time	"Return time" is the latency to return to the selected cell or region and is a measure of attractor strength. Return time is the average duration of the intervals between visits to the selected cell or region. Low return times indicate a strong attractor; high return times indicate a weak or nonexistent attractor. A region return is defined as any sequence of events outside the region beginning with an exit from the region and ending with a reentry. The number of region returns is equal to the number of region visits minus one. Returns with a duration greater than Maximum Return Time (see "Preferences") are considered to have a duration of Maximum Return Time for the purpose of this measure. The event sequence from the final region exit in each trajectory until the end of that trajectory is considered a "pseudo-return": it is included in this mean (truncated by Maximum Return Time if applicable) if and only if it has a duration greater than the mean of the trajectory's return times as calculated from proper returns alone. Otherwise, it does not contribute. The Mean Region Return Time for a trajectory group is simply the mean, across the trajectories, of each trajectory's return time. Trajectories containing fewer than Minimum Number of Returns (including the final pseudoreturn if applicable), however, do not contribute to

	the Mean Region Return Time. Events within a region of duration less than Minimum Event Duration are not taken as sufficient to terminate a return to the region, but can contribute to the return times of other regions.
Mean Return Visits	The total number of discrete visits to any of the other cells outside the selected region before returning, averaged across trajectories. Return Visits are calculated in the same as Rturn Time but with cell Visits instead of Duration as the values.
First Entry	The mean, across trajectories, of the time until the first entry into the cell or region. Trajectories that never enter the cell or region are given the maximum value, which is the total duration of that trajectory.
Last Exit	The mean, across trajectories, of the time of the last transition out of the cell or region. Trajectories that never enter the cell or region are given a value of missing.
Transitional Propensity (version 1.15a)	Using the shortest duration value found in any of the trajectories in the project to create equal-sized time bins, Transitional Propensity is a conditional probability of making the transition from time bin t to time bin $t+1$. Thus, the user must identify the cell/region of *Origin* (i.e, antecedent) and the cell/region of *Destination* (i.e., consequence), so that Transitional Propensity can be calculated by the following formula: $$\frac{\text{\# of transitions from } \textit{Origin} \text{ to } \textit{Destination}}{(\text{Duration in } \textit{Origin})/(\text{Transitional Bin Size})}$$
Transitional Entropy (version 1.15a)	Entropy calculated by the Shannon and Weaver (1949) formula: $$TP * \ln\left(\frac{1}{TP}\right)$$ Where *TP* is the Transitional Propensity value described above.

Return Time and Return Visits are based on the concept of an attractor. Shorter intervals between visits to a region indicate that the region may be an attractor, "pulling" the trajectory into its basin. As described further in Chap. 5, these measures are most appropriate when (a) returns are frequent and occur in most if not all of the trajectories or (b) attractor regions have already been identified, typically in the context of person-centered analyses. Otherwise, there may be problems with the distribution of these variables due to low base rates. Preference settings for the Return variables (see "Preferences" below) should be determined via reasonable rationale of how many returns to a region are needed to justify calculating the return measures and, more importantly, how long the trajectory can remain outside of the region before a return is considered meaningful. For example, a parent–child dyad that returns to an angry state after several hours may not be meaningful, but returns of less than 5 minutes might be.

Transitional Measures

The more recent version of GridWare (1.15a) includes a set of Transitional Measures to quantify state-to-state transitions. Because of the multidimensional and duration-based nature of state space grids in GridWare, simple transitional probabilities are not provided (though these are easily calculated from the exported values using other statistical software). Instead, GridWare provides measures based on the concept of transitional propensities derived from Event History Analysis (Mills 2011; see Chap. 7). Each moment in time is an opportunity for a system to transition to a different state. Similar to other conditional probabilities, the propensity to transition is simply a count of the number of state transitions divided by the number of opportunities to make that transition. In GridWare, the number of opportunities is derived from the event with the shortest duration in any trajectory in the whole project. For example, if the longest trajectory was 100 s and the time bin size equaled the shortest event of 0.5 s in trajectory #5, then there would be 200 opportunities for transitions to occur. For this reason, Transitional Propensity values are typically very small, but the distribution is sufficient for analysis. See Box 3.5 for more details about these measures.

Missing Values

Missing values are acceptable but these should be used with caution. For any given event, if one or both of the state values are missing, then no node is plotted for that event. However, the trajectory is linked via dotted line from the previous valid event to the next valid event. Furthermore, which missing value symbol to use can have an impact on subsequent exporting and analyses (see Chap. 4 and "Preferences" below). An alternative method for missing data is to make the missing value a le-

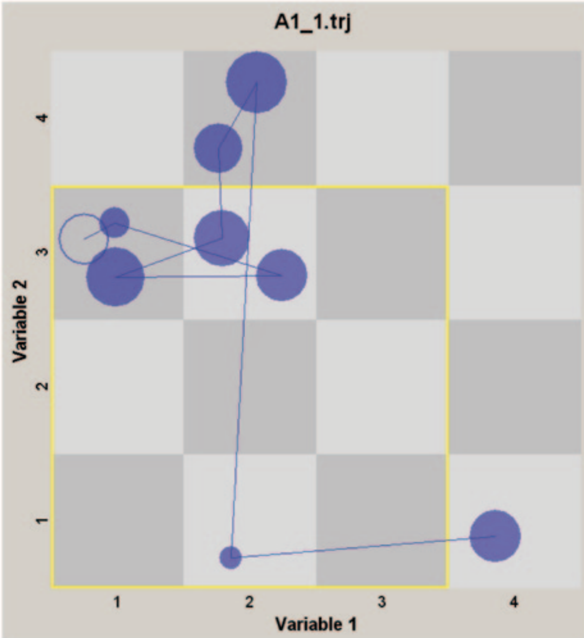

Fig. 3.6 Example trajectory with a 3 × 3 region of the state space selected

gitimate State Variable category so that there is a corresponding row and/or column on the state space grid. With this approach, the missing is treated like an "unknown" state and can be used in measures. In Example 1 (Fig. 3.3), suppose that 4 = missing. For both dimensions then, anytime a trajectory was in category 4, it meant that there was no real measurable information. Transitions in and out of these missing states could be visualized and calculated. In contrast to the first method described above, it would also allow for one dimension to be in a nonmissing state while the other was missing. Finally, it is still possible with this approach to restrict the measures to only nonmissing states by selecting the nine cells that make up the 3 × 3 region that excludes states with a value of 4 on either Variable 1 or Variable 2. As shown in Fig. 3.6, this would mean that the Whole-grid Variables would include the missing values but the Cell or Region Variables would not. In this example, comparing the whole 4 × 4 grid to the 3 × 3 region, Dispersion drops from 8 to 6 and Events drop from 7 to 5.

Preferences

The Preference Window gives the user the option to change minimum and maximum values for several calculations of measures. These include the Minimum Number of Returns, Maximum Return Time, Maximum Return Visits, Minimum Event Duration, Minimum Cell Duration, and the missing value symbol (see Fig. 3.7 for

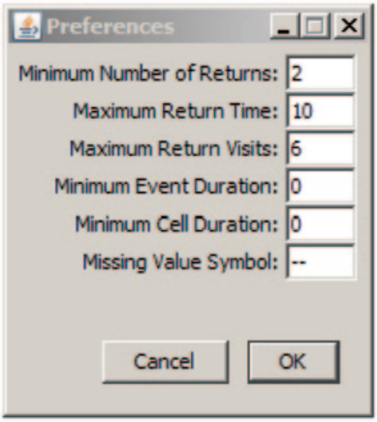

Fig. 3.7 GridWare preferences window

default values). These values should be considered carefully by the user as they can have significant effects on the values of many measures. The settings for the Return variables are discussed above. Events of duration less than the Minimum Event Duration do not contribute to: Mean Duration, Mean # of Events, Mean # of Visits, Mean Cell Range, Overall Cell Range, Duration per Event, Duration per Visit, and Duration per Cell. Cells with cumulative durations less than the Minimum Cell Duration do not contribute to: Mean Cell Range, Overall Cell Range, and Duration per Cell. The missing value symbol can be any character or set of characters. Again, the user should choose these wisely. A string character such as the default "—" or "*" can cause other software to read in the date exported from GridWare as string, rather than numeric, variables. Also important, the missing value symbol must be consistently the same across all Trajectory Files and in the Trajectories List. I will return to issues of preferences in relation to exporting variables in Chap. 5.

Exporting Measures

Measures are exported as a tab-delimited text file via the File menu on the Grid Window. The exported file is a matrix of rows (Trajectories) and columns (Variables). The first columns of this file are all of the Filter Variables found in the Trajectories List in the GridWare File. Labels for Whole-grid Measures are prefixed by "*grid*" (e.g., gridVisits, gridDispersion), Region Measures are prefixed by "*reg*" (e.g., regVisits, regDispersion) in version 1.1b and by "cell" (e.g., cellVisits) in version 1.15a, individual cell measures are prefixed by the axis and cell number in coordinate fashion (e.g., x1y1Visits, x1y2Visits), and the Transitional Measures are prefixed by the Origin and Destination cells. Most often, these variable labels need to be edited for effective use with statistical software.

The Measures Export Options dialogue box allows the user to select which measures and what groupings to export. Because of differences between the two ver-

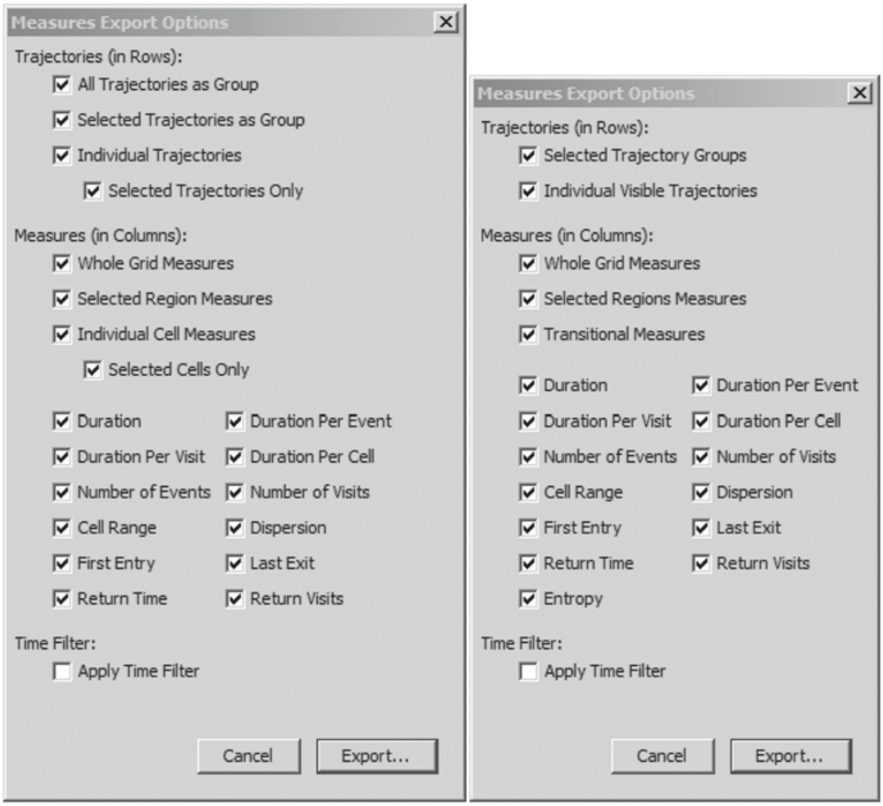

Fig. 3.8 Measures export windows from versions 1.1b (*left*) and 1.15a (*right*)

sions, both are displayed in Fig. 3.8. These dialogue boxes are separated into three sections: trajectory selection, measure types, and specific measures. Most often, users want to export one row (i.e., case) of data for each trajectory, starting with the Filter Variables found in the Trajectories List section of the GridWare File. This can be achieved by selecting only Individual Trajectories (1.1b) or Individual Visible Trajectories (1.15a) and deselecting all of the other check boxes in that section. Selecting any other options will restrict the rows of data that are exported to only those that are selected (e.g., Selected Trajectories Only) or one row of data that provides the mean for the group or trajectories of the entire project.

The next section allows you to choose which kinds of variables to export. You can export all of these at once, but that is not recommended because it can lead to unwieldy files and greater challenge in renaming the variables for analysis. Version 1.1b allows you to export measures for each and every cell in the grid via the Individual Cell Measures option. With the 4×4 grid in Example 1, that option would provide 16 variables (one per cell) for each variable selected in the bottom section of the window (e.g., 16 Visit variables, 16 Duration variables, etc.). If no region is selected on the grid prior to export, then the region measure variables will have a column but all the values will be null or missing. In version 1.15a, there is the option to select the Transitional Measures.

Finally, the bottom section allows you choose which measures to export. With Example 1, if you selected only Visits, for example, but had kept all the boxes checked for Whole-grid, Region, and Individual Cell Measures, then you would be saving a file with one gridVisits, one regVisits, and 16 individual cell visits variables (x1y1Visits, x1y2Visits, and so on). I find it most useful to export one variable type at a time and only the variables you need, rather than taking a kitchen sink approach. It is relatively simple to go back and export a new set of variables and save as a new text file.

There is also a check box for Apply Time Filter, which is left unchecked by default. This can be helpful if you want to export measures for only a portion of the time of the trajectories. I have used this, for example, to export variables pertaining to the first half and then the second half of a parent-child interaction separately (Hollenstein and Lewis 2006). To use this feature, make sure the Time Slider on the Filters Window is set to the desired range before starting the exporting process.

Run Log Window

The Run Log window displays the Java messages related to the processes and functions performed in the running of GridWare. The contents are the same as the gw_run_log.txt that is automatically created in the same directory as the GridWare program each time GridWare is opened. When GridWare opens and runs normally, it is not necessary to consult this window. However, if there is an error or problem loading in the Trajectory Files, then the user should open the gw_run_log.txt file as an aid to troubleshooting the problem. The language in the window is not entirely user friendly, but most of the time even a basic user can discern the nature of the problem. The most common errors reported in this window/file are: (1) path not found, indicating that the file and folder correspondence is incorrect or a file is missing, (2) variables in the GridWare File are not declared or incorrect, (3) the "parsing" (reading in of trajectory files) stops at the file with the problem—open that trajectory file in your text editor to diagnose further, and (4) out of memory error—the project is too large for your RAM, so it needs to be broken up into smaller groups of trajectories. Even as an expert user, I still get errors from time to time when I forget some small detail or discover a problem with one of the files I am using. The gw_run_log.txt file is very helpful for being able to diagnose and fix these problems efficiently. You will no doubt experience moments of exhilaration when after many attempts the run log reads as normal and the program opens your project.

Note that for Windows users, sometimes Java does not close when there is an error that fails to open the GridWare project. To solve this, open your task manager and see if Java.exe is in the list of processes. If so, delete it (it is ok). Also, note that on some computers Java will take up 99 % of the CPU processing. If this occurs, just reopen your project from within GridWare and Java will return to normal CPU usage.

Conclusion

That is GridWare in a nutshell. It provides the basic capabilities of state space grid analysis in a relatively user-friendly fashion. Advanced users or those who are more adept with command line applications (e.g., Splus) or programming (e.g., Matlab) will certainly find other ways to conduct state space grid analyses (see Chap. 7). For most users, GridWare is more than sufficient to meet their needs. Before getting into state space grid analyses more directly in the next section, the next chapter provides guidelines and recommendations for starting your own state space grid project.

Chapter 4
Starting Your Own Project in GridWare

The goal of this chapter is for you to be able to create your own state space grid project with GridWare. First, I will start with some initial considerations about the nature of state space as it relates to your project and some guidelines for appropriate data formats. Next, I will dive into the nitty-gritty details of text files, GridWare Files, Trajectory Lists, and Trajectory Files. At the end of the chapter, I will discuss a couple of variations that may be of interest to some readers. As with Chap. 3, the reader is encouraged to consult the GridWare User Manual that comes with the program download from www.statespacegrids.org for more details.

What Is the System?

The first, broadest question one must ask upon embarking upon state space grid analysis is "what is the system I am examining?" I am as guilty as my DS-oriented counterparts in not always declaring the system of interest explicitly in published reports, but that still does not mean it was not important at the outset. Even if you use state space grids without adopting a DS orientation, it is still essential to think carefully about the dimensions and states and the meaning of changes in those states over time. Here are some questions and suggestions to consider as you develop your own project.

First, before getting into the details of categories and states, identify what the actual system is that you are examining. You can break this down in steps by first considering whether you are looking at individual or dyadic or triadic or group behavior. Then once the scale of the system is identified (e.g., individual children, spouses, families, classrooms), the next step is to think in terms of factors that determine the behavior of interest at that scale. This is where things can get tricky for some novices. Recall that dynamic systems have been defined as a system of elements that change over time (Thelen and Smith 1994). Then what are your elements? By habit we tend to think of elements as the things on the axes of a graph. With state space grids, however, the elements are the cells—the intersection of the categories on the axes. With parent–child interactions, for example, the system *is*

T. Hollenstein, *State Space Grids,* DOI 10.1007/978-1-4614-5007-8_4,

the dyad. It is not an additive combination of parent plus child. The elements are the joint states. Surely, each individual is *nested* in the dyad, but that means they are lower order dimensions of the system of interest, the dyad. Using Fig. 1.1, we can say that the parent–child dyad is the scale of interest at the meso level. At the micro level are the individuals themselves and at the macro level is the family within which the parent–child dyad is nested.

Once you have identified the scale of your system and the elements of interest, then you can isolate the dimensions from which those elements derive. As a static (rather than dynamic) example, each individual's personality is considered by many personality theorists to be a combination of five dimensions: openness, conscientiousness, extraversion, agreeableness, and neuroticism. That is, within a five-dimensional space there is one specific point that identifies your personality, therefore any description of you (i.e., your personality system) that does not include all the five dimensions is necessarily incomplete. In my own work on socioemotional development, I have examined parent–child systems for which the elements are all of the visible and audible behaviors they exchange while interacting. For individual adolescents, I examine the physiological, psychological, and behavioral dimensions that combine to form their emotional response system. Other systems are more directly related to measurement such as eye-gaze which corresponds to spatial dimensions. The point here is that the conceptual dimensions should be considered carefully before they are operationalized as measured dimensions.

Second, moving from conceptual to measured dimensions, each dimension of state space should have two or more categories or possible values that are mutually exclusive (i.e., cannot be at two positions along a single dimension at the same time) and exhaustive (i.e., there are no other possible values not represented on the dimension). This is not negotiable. As described in the next section, failure to adhere to these two primary criteria renders state space analyses inappropriate.

Third, what is the time scale of change? State space grid analysis can be conducted at any time scale. It is important that you can see variability in system behavior at a time scale within which meaningful changes occur. For example, analyses of mood changes would not be appropriate at the scale of milliseconds or of years. Instead, moods fluctuate within the range of seconds to weeks and analyses should occur at those time scales depending on the data and research questions.

Fourth, I strongly suggest drawing many grids on paper or a white board as you begin to identify the nature of your state space and your expectations of what trajectories would look like. If it does not make sense on paper, it would not make sense with actual data either. This is a great exercise—I still do it every time I have a new grid idea or project. You can even start with an open two-dimensional space, draw some trajectories, and then decide from that what the appropriate categorization of each dimension might be.

Fifth, because state space grids are used for dynamic analyses, the behavior along any of the dimensions must be variable for at least some of the trajectories. It makes no sense to plot a constant value. How much variability? That depends on how you are going to interpret the grid patterns. For example, a more content

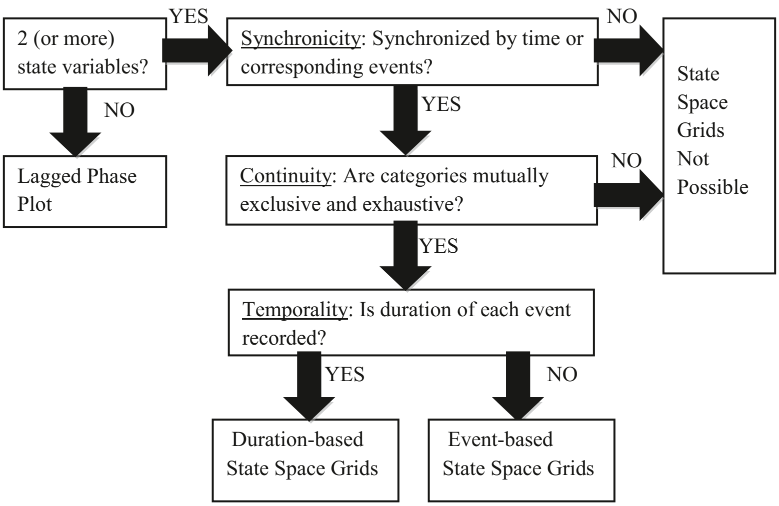

Fig. 4.1 Flowchart for determining whether data are appropriate for state space grid analysis

driven, attractor analysis may be useful with a low-variability data set, but a more structural, whole-grid analysis would not.

Considering all five of these suggestions seriously, you should be well prepared for state space grid analysis. Of course, there are further issues to consider that have to do with measurement, as I describe next, but the conceptual issues should be clear in your head beforehand.

Are My Data Appropriate for State Space Grids?

The time to start thinking about using state space grids is in the design phase of a study. There are considerations of measurement that can often be frustrating at the end if the data are not appropriate for state space grid analyses (Hollenstein 2007). The details contained in Chaps. 2 and 3 should have provided a sufficient sense of the kinds of data that have been used, but this was by no means an exhaustive report of all the possibilities. Therefore, what follows is a step-by-step method of determining the appropriateness of data for state space grid analyses following the flow chart in Fig. 4.1. Descriptions in these steps presume your familiarity with the terminology found in Box 3.2.

The ultimate goal in terms of data formatting is to obtain two linked sequences such as you would find in two columns in a spreadsheet. Each row in that spreadsheet corresponds to an "event" and the state in column 2 corresponds in time or sequence to the state in column 3 of that row (column 1 will be for onset values, see below). It is really that simple for the most basic form. I will go into greater detail about the nature of the text files used in GridWare later in this chapter. For now, it

Original			Lagged		
Onset	**Var1**		**Onset**	**Var1(t)**	**Var2(t+1)**
0	Medium		0	Medium	High
1	High		1	High	Medium
2	Medium		2	Medium	High
3	High		3	High	Medium
4	Medium		4	Medium	Low
5	Low		5		
6					

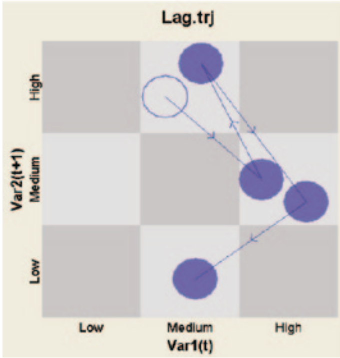

Fig. 4.2 Data and state space grid for a lagged phase plot

is only necessary that you think of column 2 as the sequence of states for the *x*-axis state variable and column 3 as the sequence of states for the *y*-axis state variable.

The first piece of information you need to determine is whether or not you have two or more state variables (spreadsheet columns). If not, all is not lost. There are ways that a single stream or sequence of values (i.e., time series) can be analyzed via a state space grid. Of course, what needs to happen is that a second axis (column 3 on the spreadsheet) needs to be created. One way would be to go back and recode your data according to some other state variable. However, there is another method that is actually often used in dynamic analyses of systems. You can create a lagged phase plot on reconstructed phase space (Heath 2000). With this method, both the *x*-axis and *y*-axis (or column 2 and column 3) are the same state variable. The difference is that the second one on the *y*-axis is lagged by time and/or event such that if the *x*-dimension is the state variable at time t, the *y*-dimension is the state variable at time $t+1$. Thus, each event (row in the spreadsheet) is plotted twice, once as the preceding event at time t (i.e., the antecedent) and once as the subsequent event at time $t+1$ (i.e., the consequence). Figure 4.2 illustrates the transformation from one state variable to the other.

Saving more elaborate detail for the section on lagged data analysis in Chap. 5, I will briefly note some features of lagged state space grids. First, note in Fig. 4.2 that the sequence becomes shorter by one event (row) because of the pairing of two adjacent events (see the section "Trajectory Files" for the reason that the final onset has no event values). Second, the lagged grid cells are now not distinct joint events but actually represent lag 1 transitions. For example, the two nodes plotted in the High/Medium cell indicate that there were two transitions from High to Medium in this trajectory. This opens up a number of possibilities, as discussed in Chap. 5. Third, similar to any lagged sequential analysis (e.g., Bakeman and Gottman 1997), the lag can be any amount of time or number of events. However, while lag 2, lag 3, and beyond are sometimes used, they are not as common as a single-step lag. Finally, once the data are in this format, all measures derived from state space grids are possible, albeit with slightly different interpretations due to the lag transitions.

Most of the time, you will have two or more state variables for a state space grid project. These variables must be synchronized. The state on the x-dimension must correspond to the state on the y-dimension through some reasonable assumption of concurrence or sequential relation (i.e., lagged variables). Ideally, this synchrony is due to measures being obtained at the same time or corresponding to the same timeline. Both Example 1 and Example 2 that come with the GridWare program are synchronized in time in this way. Events can also be “synchronized” in that they occur at about the same time or in sequence. The lagged phase plots are a good example. Other examples include bin coding in which events are recorded every 5 or 10 s or events that are only recorded when they happen (e.g., conflict bouts).

The biggest issue with data for state space grids can be the continuity of the state variables. As mentioned earlier, each state variable must be a mutually exclusive and exhaustive range of states. Mutual exclusivity means that when the system is in any one state, it cannot logically be in any of the states that are represented by other categories of the state variable. For example, if your state variable has four categories, A, B, C, and D, and it is possible for B and D to occur simultaneously, then it would be impossible to trace the sequence of states on the grid. The system cannot be in two states at once. The exhaustive criterion pertains to the comprehensiveness of the state variable. With the aforementioned four state variables, if there is a possibility that E or F could have occurred but these were not measured, then the state variable is not exhaustive.

There are some ways to work around these issues of exhaustion. One is to create a category that represents “other” states. This will allow you to account for that time in the trajectory in some way. Alternatively, you can designate those states as missing. In any case, the axis or dimension of the state space needs to include all of the possible states that could occur, even if none of the trajectories ever achieve that state. If not, it will not be possible to use the data in state space grid analysis.

The final determination of data for state space grids is to acknowledge whether states are recorded continuously in time (duration based) or is a sequence of events without information about how long each event persisted (event based). The latter is a special case of the former and it is important for the researcher to understand the limits of event-based data in terms of measures (Chap. 5). In general, a duration-based format is preferable because of the greater amount of information—the exact state of the system is known for each moment in time.

Box 4.1: How Many Cells? Sometimes a researcher has a choice of how many categories are in each dimension and therefore how many cells are on the grid. This leads to questions of what is an “appropriate” number of categories or, more cogently, what is the effect of using a smaller versus larger number of categories. First of all, there is no theoretical limit to the number of cells on a grid, but it should make sense in terms of the range of states in the actual data. As is discussed in Chap. 7, for example, we can use the number of categories flexibly to assess more continuous changes in system behavior. So the simple, yet perhaps infuriatingly ambiguous, answer is "it depends." Sometimes the state space should remain as large as it is, because there is some reason to compare to previous research or strong theoretical reasons to establish the space that way. Other times, a more bottom-up empirical approach is more appropriate for determining how many cells are in the grid.

More important, I think, is to understand the impact of such decisions. In my own work, I have used the Specific Affect (SPAFF) code system (Gottman et al. 1996) with which I have made grids for parent–child interactions. Some studies have used the complete set of specific affects (e.g., Hollenstein and Lewis 2006) while others have reduced these down to four conceptually equivalent categories (e.g., Hollenstein et al. 2004). In order to understand the impact of these choices, I have run tests comparing the raw ten-code system (contempt, anger, whine/complain, fear, sadness, neutral, interest, humor, joy, and affection) on 10×10 grids with two collapsed solutions of six categories (contempt/anger, whine/complain, fear/sadness, neutral, interest, and positive) or four categories (negative engagement, negative disengagement, neutral, and positive). For whole-grid measures, the mean values are different but they are all correlated above .9. Through my examinations of this issue, I have come to see that the biggest differences would arise by collapsing categories which are often the antecedent and consequence of one-step transitions. In the example above, people tend to transition through the neutral category rather than between states within a collapsed category. Thus, very few direct transitions between contempt and anger means that collapsing those two affective states has little impact on the calculations. Collapsing neutral and interest into one category would have a tremendous impact because of the high number of transitions between those two states.

In general, I advocate retaining cell categories as close to the original measure as possible, especially for initial analyses. Doing so would have little impact on attractor and content-based analyses because of the flexibility in choosing the cells that make up a region. The main reasons for collapsing cell categories are to minimize the complexity of presentation or because of a lack of theoretical distinction between discrete categories (e.g., anger and contempt).

Starting a New Project

To start a new GridWare project, you need to create a GridWare File to create the state space and at least one Trajectory File of the sequence to be plotted on that state space. You can create these files from scratch using your text editor (see Box 4.2). For the GridWare File, I suggest opening one of the example GridWare Files that come with the download of the program and edit it to fit your project (see below). The Trajectory Files may be easier to create in Excel or another program as well. As wtih any computer program, GridWare is relentlessly literal. Variable names and all other values must be *exactly* the same throughout. Make sure to be more vigilant than you think you should be and you should be fine.

Box 4.2: Editing Text If you are not used to editing data files as a raw text, you should know it is not similar to word processing. In fact, the first thing you should know is: DO NOT USE WORD OR OTHER WORD PROCESSING PROGRAMS to edit files related to GridWare. There are many formatting and other processes going on in the background that can render your "text" files useless for state space grid analyses. Instead, use only the text editors that come with your computer (e.g., Notepad or Wordpad on PCs and BBedit or Textedit on Macs) or one of the many text editors available from the internet. I have used TextPad (www.textpad.com) for 15 years and I can strongly recommend it. The advantage of a good text editor is being able to perform functions such as search-and-replace on multiple files simultaneously, selecting any subset of rows and columns, combining states, and many other features that make your efforts with text data much easier. I also find that manipulating data files with Excel macros is a powerful tool when first setting up a GridWare project.

Under the hood, as it were, there are several different types of encoding formats. By far, the most common is ASCII, but there is also Unicode and ANSI and others. By default, most of the time you will just be working with ASCII text and your editor will save files as ASCII. However, there are some programs and circumstances in which this may be altered. The important point is to double check that your text files are saved in an appropriate format.

All files in GridWare are tab-delimited text files. These are files that use a tab to delineate separations between columns of data. Thus, tab-delimited text is the most fluid way to move back and forth between software programs such as Excel, SPSS, and others. One advantage of tabs is that text within between tabs can include spaces, which can aid in using clear naming conventions for states, etc. (e.g., "Father Emotion Regulation Attempts" versus "FERA"). For the novice user, there are some things that may seem strange. The most obvious issue is that tabs do not form nice even columns when viewed in a text editor. This is because the number of characters between tabs can vary quite a bit. You can adjust for this in several ways. One is you can edit your files in Excel and then save them as tab-delimited text files again when you are done. This is cumbersome but functional. Alternatively, I suggest you find how to display the formatting symbols in your text editor (often this is indicated by the paragraph symbol: ¶). This will be really helpful for troubleshooting problems as well. Often the reason why a project would not open is because there is one space in a spot where a tab should be.

Fig. 4.3 The trajectory file A1_1.trj from Example 1

Onset	Variable 1	Variable 2	Variable 3
0.00	1	3	2
1.65	1	3	3
2.30	2	3	3
4.10	1	3	3
6.48	2	3	3
8.60	2	4	3
10.25	2	4	1
12.82	2	1	1
13.17	4	1	1
15.00			

All data for a project must be saved in the same directory, although this does not have to be the same folder as the GridWare program. The folder that contains the data (Trajectory Files) must be in the same directory as the GridWare File and named *exactly* by the combination of the GridWare File filename followed by "_ trjs". For example, if the name of your GridWare File is "Cohort 1.gwf", then the data to which "Cohort 1.gwf" refers must be in a folder named "Cohort 1_trjs". Below I will describe the necessary features of Trajectory Files and the GridWare File in turn.

Trajectory Files

The structure of each Trajectory File must conform to the same format according to the following rules: (1) The first row contains the labels for each column; (2) There must be only one column for Onset (column 1), and at least 2 other columns of state variables; (3) Each row is a new event. For event-based data where each new row is not an onset time but a new event, Onset can be simply an ordered sequence of integers; (4) The final row must have only an Onset value, but no values or spaces for any other variable. That is, the last digit of the last onset time (end time) must be the end of the file. This value represents the Offset of the final event whose states are identified in the previous row; (5) Separation between columns must be tab delimited. Blank spaces are considered as the part of a single text label or value; (6) Onset times can be in one of several formats: decimal (…0000.000…), real time (hh:mm:ss), or a combination (mm:ss.00). All onsets will be converted to seconds, however, both within the onscreen display and in any file output, including when resaving trajectory files; (7) Missing events must be represented with a symbol that is consistent with the user-defined symbol set in the Preferences menu option (see Chap. 5 for recommendations for missing value symbols). For example, Fig. 4.3 shows the short trajectory file (A1_1.trj) with three state variables that comes with the GridWare download in Example 1. In this file, the trajectory starts in Variable 1 = 1, Variable 2 = 3, and Variable 3 = 2. The next event is 1.65 s later when Variable 3 switches to 3, but the other two variables remain the same as before, so the display does not change. At 2.3 s, Variable 1 now shifts from category 1 to category 2. The trajectory ends at 15 s. Note, how this final offset does not have any values in the columns to the right. This is necessary for every Trajectory File and often a source of error for novice users who forget to arrange the data in that way.

Creating Trajectory Files can be tedious without some way to automate the process. Often, this will be easiest with some light programming via a scripting language or Excel macros using Visual Basic. I have begun to facilitate that by providing tools on the www.statespacegrids.org website. The GridWare File Converter is another Java application that is in the beta-testing phase. With it, you can convert Noldus Observer Files (ODF files up to version 6.0) into a trajectory file format. Over the next few years there will be more options for file conversion. Once you have your project into trajectory files, the GridWare File Converter also allows you to modify state variables. This comes in handy for lagged analyses and combinations for three-dimensional state space grids. State variables in all of your trajectory files can be simultaneously concatenated, summed, subtracted, multiplied, deleted, renamed, lagged, or concatenated. This is a handy tool for even the novice user.

GridWare File

Figure 4.4 shows the GridWare File for Example 1. GridWare Files are configured in a similar way as an html file in that the start of a section is marked by a tag in arrow brackets (e.g., <GridWare>) and the completion of that section is marked by the same thing with an added slash (e.g., </GridWare>). Do not change these as they are required for Java to run the program. There are two main sections, the configuration section that defines the variables and preferences (<config> to </config>) and the trajectories list section that defines which trajectories are included in the project (<trajectories> to </trajectories>). Within these two main sections are four subsections that provide essential information specific to the project.

The first subsection is the declaration of the filter (aka trajectory) variables. These are the unique between-trajectory variables that identify each trajectory. You need at least one but can have as many as you want. The primary variable is usually some sort of identifying variable such as Subject ID #. In this example, it is Family. Across the versions of GridWare, there are several ways to declare these variables in the first column: trajectory, identifier, or attribute. All three are equally acceptable. The next column declares the type of variable: ordinal (or integer) or categorical. For ordinal filter variables it is necessary to declare the minimum and maximum so that the sliders that are created in GridWare have accurate anchor points (e.g., variables wave and age in Fig. 4.4). Alternatively, filter

```
<GridWare>
<Config>
identifier      categorical     family  A1      B2      C3
identifier      integer wave    1       3
attribute       integer age     9       13
attribute       categorical     gender  male    female
state   integer Variable 1      1       4       1       2       3       4
state   integer Variable 2      1       4       1       2       3       4
state   ordinal Variable 3      1       4       1       2       3       4
MinReturns      2
MaxReturnTime           10
MaxReturnVisits         6
MinEventDuration        0
MinCellDuration         0
MissingValueSymbol --
</Config>
<Trajectories>
Family  Wave    Age     Gender  Filename
A1      1       11      male    A1_1.trj
A1      2       12      male    A1_2.trj
A1      3       13      male    A1_3.trj
B2      1       9       female  B2_1.trj
B2      2       10      female  B2_2.trj
B2      3       11      female  B2_3.trj
C3      1       10      male    C3_1.trj
C3      2       11      male    C3_2.trj
C3      3       12      male    C3_3.trj
</Trajectories>
</GridWare>
```

Identifiers/Attributes = Filter or Trajectory Variables

State Variables

Preferences

Trajectories List

Fig. 4.4 The GridWare file from Example 1

variables can be categorical. This just means that they will show up as a list in the filters window. Declaring categorical filter variables only requires the name of the variable to be declared.

The next subsection declares the state variables that are found in the trajectory files, which are always declared as state variables in the first column. The declaration of ordinal versus categorical is slightly different than with filter variables. This is because we are declaring the axes of the state space now, and the categories have to be unambiguous. For categorical state variables, the name must be declared followed by each of the possible values in the order in which you want them to appear on the grid dimensions (note that if you want to rearrange the grid categories this is one way you can do it quickly and simply). For ordinal state variables, you must declare the minimum and maximum followed by each discrete value. Note that these can also be declared as integer variables. There is really no functional difference between categorical or ordinal state variables. The difference is really about circumstances when your state variables in the Trajectory Files are numeric but you may want to re-label the numbers with text labels. This gives you the flexibility to rename your integer values in the GridWare File without having to change them in every Trajectory File.

The next subsection is the Preferences section. You actually do not need to include this section because GridWare will revert to the default values (the same values in Fig. 4.4). If you want to control these, then this is the best time to do so, as you are setting up your project. The implications of preferences are discussed with the measures in Chap. 5.

The final subsection is the trajectories list. This section is a matrix of variables and cases similar to a statistical spreadsheet. Every variable column in the trajectories list—other than the Filename column—must be declared in the trajectory variables declaration at the top of the GridWare File (Fig. 4.4). No extra variables may be declared in either section. These filter variables can have missing values. However, these should be infrequent as it seems that having too many missing values for filter variables causes the program to react unexpectedly.

Opening a GridWare Project

To open a project, first open the program (double click GridWare 1.1b.exe or GridWare 1.15a.jar). Then browse to select the GridWare File (e.g., Example 1.gwf) for the project you want to open. When you first open a project, nothing is displayed on the grid. To display trajectories select the filters, the display nodes, and display transitions options.

Saving a Project

You can close down GridWare anytime you are finished with a project and do not need to save it. The GridWare File and Trajectory Files will remain just as you created them. If, however, you want to save new state variable categories (because you altered them on the *x*-axis), the color of the nodes (because you recolored them), or the position of the nodes in the cells (because you rerandomized them), then select save. GridWare will save over the existing project unless you rename it to something else. I strongly recommend you save it as another project if you have made any of these changes. You never know when you might need to go back to the original.

Conclusions

With the steps in this chapter you will be able to create your own GridWare project. As suggested earlier, the best approach is to use preexisting GridWare projects as a template to edit rather than starting completely from scratch. A good text editor, attention to detail, and appropriate data, you should be fine. That being said, even I, the de facto expert, have to go through several iterations of trial-and-error troubleshooting when starting new projects. The program will discover anomalies buried deep in one of the trajectories. Use the gridware_run_log.txt to decipher the error messages and fix any problems that arise (see the section "Run Log Window" in Chap. 3). Once the project is open, have fun exploring the data!

Chapter 5
Within-Grid Analyses

Once you have successfully created your project in GridWare, there are now a range of analyses that you can execute. The analyses described in this chapter all pertain to analyses on the indices obtained within each grid, one for each case (row) in the trajectories list. In contrast, the next chapter, Chap. 6, will cover the analyses in which grids are compared with each other, most often to answer within-subjects developmental questions. With the basic understanding of these two chapters in this section, you will be well prepared to consider the more advanced analyses that can be conducted with state space grids described in Chap. 7.

Analyzing Exported Data

As described in Chap. 3, you can export any number of variables from GridWare as a tab-delimited text file. The cases (rows) will be the individual trajectories found in the rows of the trajectories list. The columns are the variables that you have selected for output, starting with the filter variables. These files are essentially text spreadsheets that can be mostly imported into any database or statistics software program. Because of their widespread use and my experience, I will focus on the combined use of a text editor, Excel, and SPSS. In fact, in the preparation and analysis of projects in GridWare, I almost always have those three programs open simultaneously.

For the first set of analyses described below, it will be best to work together on an example. Open up the Example 1.gwf project that comes with the GridWare download. Select the four-cell region formed by the combination of category 1 and 2 on Variable 1 and category 1 and 2 on Variable 2 so that it looks similar to the state space grid in Fig. 5.1. Next, export both whole-grid and region measures for each trajectory by selecting the export options as shown in Fig. 5.1. Once complete, open the exported file from wherever you saved it in your text editor. Now open the Example 1.gwf file in your text editor as well. Comparing the two, the first five columns of the exported data file should be exactly the same as the columns in the "Trajectories List" section of the GridWare File. All subsequent columns to the right are values calculated and exported by GridWare. Scroll to the right and you will see

T. Hollenstein, *State Space Grids,* DOI 10.1007/978-1-4614-5007-8_5,

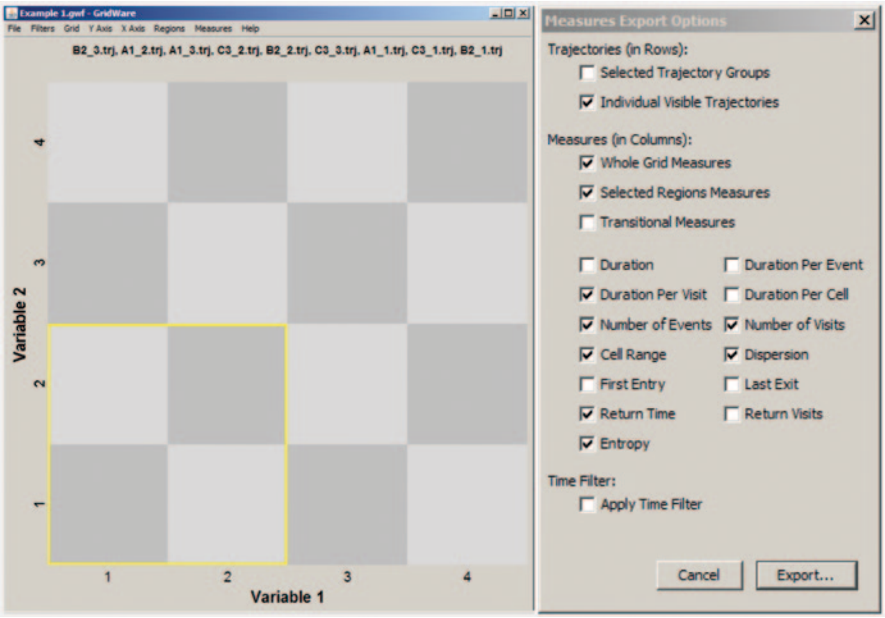

Fig. 5.1 GridWare window with a four-cell region selected and measures export dialogue box for Example 1 data export

numbers or the missing value symbols (−) for every case-variable (row–column) combination. You will also notice that the variable names continue on well beyond the data—this is simply due to the differing number of characters between the tabs. To make this file easier to read, import it into Excel.

By default, Excel will have the file type set to Excel files. You must change this to "All Files" or "Text Files" in order to see your exported data file. Once selected, Excel automatically brings you through the Text Import Wizard. Make sure that you select the following: Delimited button, Delimiters = Tab only, and General Format. Earlier versions may have the option for headers (variable names in the first row) and if so make sure that is selected as well. Now the exported data is in Excel for easy viewing. The first new variables to the right of the filter variables are numTrajs and gridCells (columns F and G), which are exported by default every time. With the export of all individual trajectories (rather than exporting trajectories as a group—the check box we deselected in the Export Options window), these variables are not very informative because they are constant. However, if you are doing a number of exports from the same project, the number of gridCells may be useful information. The next three columns are also only useful in a limited number of situations. The variable gridDuration gives the difference between the first onset and final offset—the total duration—for each trajectory, which in this example is the same for every trajectory. There are no missing data in this project, so the values of missingDuration and missingEvents are all zero. These variables can be very helpful if there are missing values, especially if the amount of missing is not yet known to you.

I strongly recommend saving files in each format (text, Excel, SpSS, etc.) as you prepare your analyses. It will save you time later if you have to go back and redo any step of the process. Save the Excel file now.

To import these data into SPSS, you will follow a similar but longer import procedure. You could, of course, just import the Excel file but for illustration purposes (and because you may want to skip the Excel step) we will import the text file directly. Before importing files in SPSS, you must make sure that neither the text nor Excel files are open in other programs. For SPSS, select the following in the Text Import Wizard: No predefined format, delimited, yes variable names are included (often need to change this one from default), first case on line 2, each line represents a case, import all cases, only tab—not space—appears as a delimiter between variables (often need to change this one from default), no text qualifier, click ok if it gives the error message that invalid variable names have been found (they are fine but we will cover this later), just click next for the variable specifications, and then finish. After checking the data view to make sure it all worked as expected, switch to the variable view. Here you will be able to detect any problems with your variable types. Specifically, those variables that are the result of division (DurationPerCell, DurationPerEvent, and DurationPerVisit) as well as region variables (especially ReturnTime an ReturnVisits) are most likely to have missing values. The division problem arises whenever there is a division by zero, resulting in a missing value. The other region measures are also affected by the lack of any visits to the selected region. Dispersion in a region, as you can see in the data file, is 0 if only one cell within the region is visited and missing if never visits. The return variables are further constrained by settings in the Preferences, as described in Chap. 3.

The net result is that you must be very careful and conscious of the issues of missing values and variable types in your exported data. First, one of the most important things is to understand the difference between zero and missing. When possible, this distinction should be retained and GridWare exports have been designed to facilitate that distinction whenever possible. Still, it may be better in some instances, for example, to recode all missing values for DurationPerVisit to zero as long as there was a valid value of zero Visits. First, Missing values should be exclusively for cases where there is no information. Second, carefully inspect your case counts (n) in any analyses as a double check of the variables you are analyzing. Third, because the default missing value symbol is text, SPSS will automatically make that variable a string variable. This is a pain but there are several ways around it. I have found that the simplest way is to recode the missing value symbol earlier in the process. You could change the missing value symbol to a "." or "99999" in GridWare preferences before exporting (SPSS uses the period for its system missing value symbol). However, if you take this step, it will also have to be consistent with the missing symbol in the Trajectory Files. If you edit the GridWare File file in a text editor, you will also have to change all of the individual Trajectory Files. If you change preferences from within the program, then it will change all of the files for you. Alternatively, you can change the missing value symbol via search and replace options in your text editor or Excel. Search and replace is a powerful tool so be very careful. It is very easy to alter unintended values. The most cumbersome solution is to recode the variables in SPSS. To do this, you would first have to recode the missing symbol to a number (e.g., 99999) and then convert the variable from string to numeric. With only a few variables, this is not a challenge, but with many variables and many cases, it can be daunting.

Exporting variables from GridWare for use in other programs is reasonably straightforward, but you need to be vigilant and systematic. Do not rush the process and check the results of each step carefully. I cannot emphasize that enough. Impatience and lack of focus invariably lead to errors or, at the very least, detecting issues later in the process that can only be resolved by starting over from the beginning. I also recommend that you do not stop in the middle of the process to finish at a later time. Go from GridWare to analyzing the descriptives in SPSS in one sitting. You have been warned….

Now we can consider different types of analyses that can be conducted. I will begin with the most basic and get progressively more elaborate as the sections progress. This set of analyses is not necessarily exhaustive (see Chap. 7) but it provides a good overview of the most central to state space grids.

Whole-Grid Analysis

Some research questions pertain to the overall pattern depicted on the state space. In general, the quantification of these patterns falls under the heading of variability with several different aspects of variability derived from the measures available. The primary measures of variability are cell range (mean and overall), dispersion, visits, duration per visit, and entropy. There are also simple calculations that can be made using the basic variables exported by GridWare such as Transitions per Minute and the Event/Visit ratio. Readers are encouraged to consult the studies that have used these measures in order to get a full understanding of their utility (Granic et al. 2003, 2007, 2012; Hollenstein 2007; Hollenstein et al. 2004; Hollenstein and Lewis 2006; Lunkenheimer et al. 2011; Martin et al. 2005). There is really nothing complicated about whole-grid analysis as it essentially amounts to exporting the values for use in statistical models. Here, I will just review some of the conceptual and computational properties of these measures that relate to their analytical interpretation.

Cell Range

The count of the total number of cells occupied is perhaps the most basic measure to be derived from state space grids. It is also the most content driven and least dynamic of all the whole-grid measures. A greater number of cells necessarily means a greater range of content. For this reason, the conceptual interpretation of the cell count is the *range* or repertoire of the system. Cell Range is a somewhat static snapshot because it provides none of the details of that frequency and duration information can provide. Still, it correlates moderately with other variability measures and has shown convergent validity in several studies (Granic et al. 2003; Hollenstein et al. 2004). One variation on Cell Range is to divide by the total number of cells on the grid to get a proportion score. This proportion can be useful if there is a need to compare the Cell Range of two or more grids of unequal size.

Dispersion

Cell Range equates a single brief visit to a cell with long and/or repeated visits to a cell. Dispersion was devised in order to compensate for those kinds of biases in base rate durations. You can think of Dispersion as Cell Range controlling for proportional durations in each cell. Thus, Dispersion is a relative value that depends on the number of cells in the state space grid as well as the general distribution of cell visits. For some projects, it may be most typical for any given trajectory to occupy only half of the cells; in other projects, each trajectory visits most or all of the cells at least once. Dispersion values would be higher in the latter case than in the former. For this reason, Dispersion comparisons between trajectories can reveal meaningful differences within a project, but comparison of Dispersion values across projects that differ in the dimension categories are not meaningful. With any project that has duration information for events, Dispersion is probably preferable to Cell Range and is why it has been used more consistently in more recent studies (e.g., Granic et al. 2007; Hollenstein and Lewis 2006; Lunkenheimer et al., 2011).

Visits and Transitions

Each trajectory begins in one cell and as time progresses tends to visit other cells on the grid. Thus, Visits are really a count of transitions but are always off from the actual number of transitions because the first event in the first cell is counted as a visit. Transitions are equivalent to the number of visits minus 1. Transitions are the most content-independent and dynamic measure of variability because transitions can occur between any number of cells regardless of content. For instance, a count of 40 transitions in a given trajectory could have occurred between two cells or across 39 cells. For this reason, it may be useful to conceptualize transitions as a measure of the propensity to switch states or dynamic flexibility (Hollenstein et al. 2012).

Frequency-based measures such as Visits, Transitions, and Events are somewhat dependent on time. The longer the observation or time series, the more opportunities there are for new events and transitions. For example, in some studies the duration of parent-child discussions has varied by as much as a minute within a given task or as much as 5 min across tasks (e.g., short warm up followed by longer problem solving discussion). Therefore, I recommend that in all but the most controlled instances (i.e., duration of each trajectory is exactly the same), that the frequency measures be divided by the total duration of the trajectory. To date, this has most often resulted in the use of the variable Transitions Per Minute, calculated as (gridVisits − 1)/(gridDuration/60), for projects with seconds as the time units (Hollenstein et al. 2004; Hollenstein and Lewis 2006). An alternative is to divide the trajectory into equal duration segments (e.g., four 2-min segments of a task lasting at least 8 min) and analyze accordingly (see Chap. 7).

Duration Per Visit (Average Mean Duration)

We can calculate a mean duration in each cell by taking the total duration in that cell and dividing by the frequency of visits to that cell. This gives an indication of how strongly the trajectory got "stuck" in that state. By extension, we can calculate the overall "stuckness" or rigidity of a trajectory by taking the average of all of the individual cell mean durations. This average mean duration is calculated simply by dividing the total duration by the number of visits. This rigidity is the converse of the dynamic flexibility of Transitions, reflected by the typically strong negative correlations between these measures. The difference is that Duration Per Visit is a duration-based measure and thus contains slightly different information.

Visits/Events (or Events/Visits) Ratio

One variation on the basic frequency of transitions derived from the Events variable is to calculate it in terms of the number of Visits. With no repeating events, this ratio is 1, with many repeating events this ratio approaches 0. This ratio was first used to compensate for a lack of duration information in an event-based project. Developed by Carol Martin, the composite index of 1—Visits/Events quantified the "stickiness" of a cell or region for event-based data (Martin et al. 2005). Thus, it functions in the same way as Duration Per Visit as indicator of rigidity in projects where duration is not known. An alternate variation is the Events/Visits ratio. For this variable, a ratio of 1 is the lowest possible value and reference point. In circumstances where repeating events are not possible, then by definition Visits will equal Events and this ratio is a useless measure because all trajectories will have a value of 1. With repeating events, however, the degree to which Events exceed Visits indicates "stuckness" or "rigidity."

Entropy

Entropy for state space grids was derived from early information theory (Shannon and Weaver 1949) as a way to quantify the orderliness or predictability of communication sequences. Tom Dishion added this measure to the state space grid toolbox in order to capture the predictability of deviant peer interactions (Dishion et al. 2004). One interpretation of Entropy with this formula is the number of "bits" of information necessary to recreate the entire sequence of events. As a contrast, consider two extremes. For a sequence of events that alternate (e.g., ABABABAB), only one bit of information is needed to recreate the entire sequence; if you know A, then you know the entire sequence. This is an example of a completely ordered, maximally predictable trajectory (low Entropy). In contrast, a sequence of random events (e.g., ABAADCPZKB) would require all of the bits of the sequence in order to recreate it. This is a highly unpredictable, maximally entropic trajectory. Thus, what we obtain from Visit Entropy is a direct measure of the unpredictability of the trajectory event-to-event sequence.

While Visit Entropy is based on the conditional probabilities of event-to-event transitions, Duration Entropy is calculated from the proportional durations or duration probabilities (i.e., cell duration divided by total duration). While both use probability information within the same formula, Duration Entropy was included in GridWare simply as a logical extension of existing calculations performed by the software. Upon further testing over the past few years, it has become apparent that, in most cases, Duration Entropy and Dispersion are strongly and sometimes nearly perfectly correlated. There may be anomalous conditions for which this is not true, but my recommendation at this point is to stick with Dispersion rather than Duration Entropy. It is a more direct and interpretable measure.

I have consistently found that these whole-grid variables are normally distributed, lending themselves to mainstream linear statistical procedures. However, that does not mean you should not check these distributions before analyses. All of these measures are available for regions as well, but you should be cautious when examining the variability within a region. The variables will be censored in possible odd ways depending on how often the trajectory moves into or out of that region. The smaller the region, the more there will be issues.

Attractor Analysis

As described in Chap. 1, the three main dynamic systems (DS) concepts addressed by state space grid analysis are variability, attractors, and phase transitions. We have covered variability with the whole-grid variables described above. Now I will describe some ways to detect attractors and assess their strength. Note that I will only be describing one type of attractor, a point attractor. There are other kinds of attractors (e.g., cyclic, chaotic) that are not covered here. In general, attractor analysis should only be performed if there are a reasonably high number of events in the trajectories. Trajectories with only a few events cannot support reliable estimates of attractor location or strength.

There are two approaches to attractor analysis with state space grids. One approach is to identify attractors a priori based on theory. For example, major depression is episodic and its recurrence has led to the theoretical assertion that it is an attractor (Johnson and Nowak 2002). Similarly, Granic and Patterson (2006) predicted that coercive family processes are also attractors. Starting with the theoretical premise or prediction of an attractor, these can be modeled as cells or regions on a state space grid. Once identified, several of the region measures can be used to assess attractor strength. For example, Granic et al. (2007) broke up the parent–child state space grid into several meaningful regions and analyzed the strength of those regions for each dyadic trajectory (Fig. 2.6).

The second is a bottom-up approach to detect attractors empirically on a grid-by-grid basis. The basic rationale of this approach is that an attractor state should occur at least more often than by chance and certainly more often than most other states. There are other ways to identify attractors in a more computational fashion (e.g., null clines, see Gottman et al. 2002), but here I will only describe the methods relevant for state space grids. Once attractor cells or regions are identified, then the assessment of their strength follows the same process as with the theoretical attractor identification approach. Therefore, I will first describe the empirical identification approach before detailing the state space grid variables that can be used in either approach.

Empirically Derived Attractors

The most straightforward method of identifying an attractor state on the grid is to simply find the cell or cells with the highest mean durations, duration per visit, or visits. This is a coarse approximation casting a wide net, but it should correlate well with more rigorous methods of establishing which states are attractors. That is, with this approach you will get false positives but you should also identify most, if not all, actual attractors. I do not recommend using just these measures of frequency or duration for attractor identification (though for attractor strength these are fine). Instead, examining these measures can be a useful first step or you can report about these states as important, but not as attractors per se.

The more formal and compelling justification for an attractor state is that it is more probable than other states. Hence, we need some way of differentiating inconsequential variation from systematic variation reflecting the underlying structure of a system. With state space grid analysis, you can use the method described by Lewis et al. (1999) in the first state space grid application, the winnowing method. The procedure begins with all of the individual cell durations (or visits) and progresses in iterative fashion by eliminating the cell with the lowest duration in the set one-by-one in each step. Eventually, the cell or cells with the highest duration

Table 5.1 Derivation of attractors with the heterogeneity score in a six-cell (2 × 3 grid) example. Cells x2y2 and x2y3 are identified as an attractor region

Step	Duration							# Cells (C)	Expected (D/C)	
	x1y1	x1y2	x1y3	x2y1	x2y2	x2y3	Total (D)			
1	2	2	3	4	20	30	61	6	10.17	
2		2	3	4	20	30	59	5	11.80	
3			3	4	20	30	57	4	14.25	
4				4	20	30	54	3	18.00	
5					20	30	50	2	25.00	
6						30	30	1	30.00	
	((Observed − expected)^2)/expected						Sum	Cells	H-score	H-proportion
1	6.56	6.56	5.05	3.74	9.51	38.69	70.11	6	11.69	1.00
2		8.14	6.56	5.16	5.70	28.07	53.63	5	10.73	0.92
3			8.88	7.37	2.32	17.41	35.98	4	9.00	0.77
4				10.89	0.22	8.00	19.11	3	6.37	0.55
5					1.00	1.00	2.00	2	1.00	0.09
6						0.00	0.00	1	0.00	0.00

H *heterogeneity*

remain based on a criterion *heterogeneity score*. Table 5.1 shows the steps of the winnowing technique in a small example with cell durations in a 2 × 6 state space grid. First, the null hypothesis is that all of the behavior on the state space is equally distributed such that each cell's duration is the same. This may seem familiar to you as it is the basis of chi-square tests of independence. Thus, the Expected value in each cell is calculated as the total duration divided by the number of cells *in that step*. Next, each cell's deviation from that expected value is squared (to eliminate negative values) and summed across cells. This sum is then divided by the number of cells to obtain the heterogeneity score:

$$\text{Heterogeneity}_j = \frac{\Sigma(\text{Observed}_i - \text{Expected}_j)^2/\text{Expected}_j}{\#\text{ of Cells}_j}$$

where i is an index of the cell and j is an index of the current iteration. Thus, the heterogeneity score calculated at each iterative step in the winnowing process decreases as the number of cells included decrease.

Finally, to identify which cell or cells is an attractor, heterogeneity values are quantified as a proportion of the first heterogeneity value from the first iteration ($\text{Heterogeneity}_j/\text{Heterogeneity}_1$) and examined for scree, the value after the largest drop in proportions. If no large drop occurs ("large" was defined as approximately 50 % or more in Lewis et al. 1999), then the scree value can be taken as the last drop to 0 for the final single-cell iteration. Thus, with two or more cells identified as an attractor, there must be a large drop (> 50 %), but if only one cell is an attractor, there may or may not be a large drop. Moreover, if more than one cell is identified as an attractor, then this could be one larger attractor region (if the cells are adjacent) or more than one attractor (if they are not adjacent). In the example in Table 5.1, the drop in the heterogeneity proportion (H-proportion) from 0.55 to 0.09 is the scree indicating that two cells in this example qualify as attractors. As these are two adjacent cells, the combination is identified as a single attractor region.

Several variations on this method can be useful as well. For instance, instead of cell durations, you could use visits or duration per visit. An interesting approach would be to use return time but then cells with longer return times would be eliminated first and the goal would be to winnow down to those cells with the shortest return times. The expected value can be determined in other ways, depending on what is most logical for the states in the project. For instance, the null hypothesis may not be that behavior is equally distributed across all cells but that it should reach the sample mean.

A second way to empirically derive attractors is to use the return time variable with constraints specified in the preferences window of GridWare. Attractors should be a relatively small subset of all the cells on a grid and therefore the criteria for identifying them should be restrictive. By increasing the minimum number of visits required to calculate return time, you can eliminate cells with only a few visits. Return time will then only be calculated for those few cells or trajectories that meet that minimum criterion. There are many possible ways to determine what this minimum value should be. One approach would be to use the total number of visits (transitions) and divide by the total number of cells (e.g., visits per cell). Any cell with more visits than the visits per cell value would have a return time value computed. However, the returns to that cell may not occur quickly enough to count as a return. By adjusting the maximum return time you can further limit the number of cells that would qualify as an attractor. Returns that exceed the maximum do not get counted in the calculation. This is not a precise method but it will allow you to home in on attractor states with a reasonable amount of certainty.

Attractor Strength Measures

Once a region on the state space is identified as an attractor, the relative strength, or pull that the attractor has on the trajectory, can be quantified in several ways. However, the method of deriving these attractor regions results in distinct analytical implications. Attractor strength for a priori attractors reveals individual differences relative to the same attractor for all trajectories. On the other hand, empirically derived attractors are unique to each trajectory (attractor A for trajectory 1, attractor B for trajectory 2, and so on). Thus, comparisons across cases for idiosyncratic attractors derived empirically require person-centered analytical approaches. This can be a case-by-case approach or simply interpreting attractors in an entirely structural way, independent of the content of the attractor states, as was done in the Lewis et al. (1999) study.

The most direct way to measure attractor strength is the return time or return visits measures. Shorter return times or visits reveal that it is difficult for the trajectory to extricate itself from the region. Longer return times or visits indicate a weak attractor or even that the region is not an attractor at all (see previous section). The return variables are sometimes difficult to work with if what you are looking for is a normally distributed variable for statistical analysis. Missing values are often the lion's share of values for the return variables in the exported data files. Thus, return time and return visits are most often useful only after an attractor has been identified. That is, they are most naturally suited to person-centered and case-by-case analysis. Exceptions include projects with trajectories that visit all of the cells on the grid frequently. This can be due to the overall length of the trajectories (e.g., family observations of several hours) or the relative frequency of transitions across most states.

Another method of analyzing attractors that can be more amenable to mainstream statistical assumptions is through the first entry and last exit variables. The logic is that an attractor state should occur relatively early in a trajectory and probably be the last or one of the last states at the end of the trajectory. That is, attractors should have low first entry values and high last exit values. Note that this can get complicated if the trajectory never goes into the region (first entry would equal to the total duration and last exit would equal to 0), so use these variables cautiously. An example of how this may be useful is with the problematic state of mutual negativity in interpersonal interactions. A couple, parent–child dyad, or family with low first entry values into mutual negativity would be relatively dysfunctional compared to those with higher first entry values. One metric of treatment success in these cases might be if the first entry value increases over measurement occasions.

To really establish a comprehensive sense of the attractor landscape revealed by the trajectory patterns on the state space grid, it may be best to use a combination of methods and variables. Strong attractors will have frequent visits, long mean durations, short return times and return visits, and low first entry values. A cautious approach might be to use these variables as multiple criteria for establishing a region as an attractor.

Attractors Over Time

The Lewis et al. (1999) study tested the stability of attractors across a 4-month interval with correlations of attractor strength variables. One method of attractor analysis that has yet to be tried is to examine attractor change and stability over time *on the state space*. Using a variation on an existing study, I will illustrate this possibility. In the Lewis et al. (2004) study, each toddler had a trajectory at each month from 14 through 24 months of age. If attractor analyses were performed for each of those monthly trajectories, then the single strongest attractor cell could be identified. With the same 5×5 grids, then, a new trajectory can be made for each toddler such that each event is the x and y coordinates of the attractor for that month. In the trajectory files, onset values would start at 14 and end at 25 and there would be 11 events in every trajectory. Toddlers with stable attractors would have most or all of their nodes in one cell or cluster of cells. Toddlers with changing attractors would have more than one region where the nodes were plotted. In this way, one could conduct an examination of attractor stability as part of state space grid analyses.

Transition Analysis

Often researchers want to know information about the sequence of events, which states follow or precede other states. There is a long and rich history of lag sequential analysis using confusion matrices and conditional probabilities. These analytical techniques are quite complementary to state space grids and are discussed in more detail in Chap. 7. In this chapter, I will cover the transition analyses made possible by state space grids and unique to GridWare: Transition measures and lag grids.

In version 1.15a, GridWare provides a method for quantifying transitional propensities. These are the probabilities of transitioning from one region to another at any moment. In a sense, they are like traditional conditional probabilities—the probability of an A-to-B transition given the trajectory is in state A. However, these are different because the probabilities are calculated using time information as well. To do this, the time series is broken down into small time windows or bins. Transitional propensities come from the bin-to-bin probabilities of transition from A-to-B, given A. Thus, these are typically small values but they have reasonable distributions when at least a few transitions occur in most trajectories. To obtain these measures, you first select the cell or cells that make up your origin region (i.e., the antecedent). Then select the cell or cells that make up the destination region (i.e., the consequence). To export, simply keep the Transitional Measures box checked. Once exported, you can analyze these variables as usual.

The interest in transitions often arises in the interpersonal dynamics literature. Researchers are interested in who is "driving" an interaction or what the responses are to a particular behavior. Transitional propensities provide a precise measure of those reactions. Often, it is useful to export several different transitional propensity variables based on different consequences for the same antecedent or based on reciprocal antecedent-consequence combinations (e.g., parent to child and child to parent).

State space grids also offer another way to examine transitions between states through the use of lagged variables. In Chap. 4, I mentioned the lagged phase plot as an option for single state variable projects. These were just one example of a more general approach to have some state variables that are lagged relative to other state variables. This opens up a whole world of possibilities.

The basic idea is to take an existing state variable, copy all but the first value, and paste those values in the next available column just below the header row (see Fig. 4.2). Now the first event (row) in the trajectory file is a combination of the original variable at time t and the same variable at time $t+1$. However, with state space grids you can have more than one state variable and therefore you can have

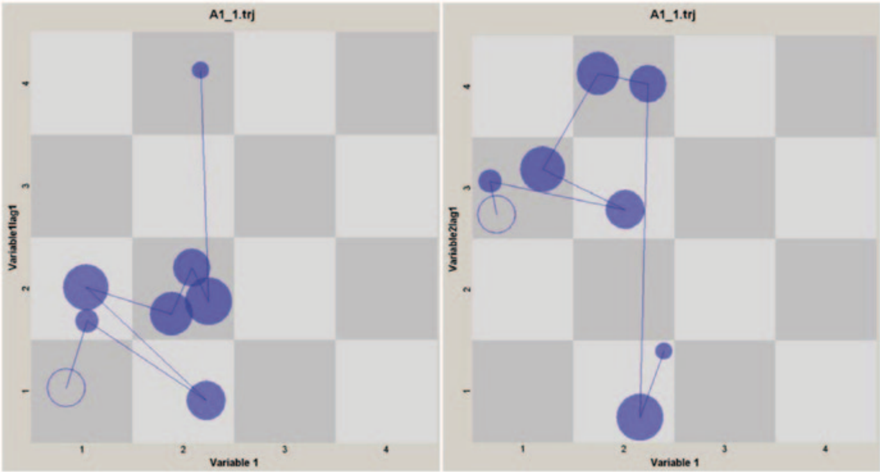

Fig. 5.2 State space grids with lagged variables from Example 1. The grid on the *left* is a lagged phase plot of Variable 1 at time t on the x-dimension and Variable 1 at Lag 1 $(t+1)$ on the y-dimension. The grid on the right is Variable 1 at time t on the x-dimension and Variable 2 at Lag 1 $(t+1)$ on the y-dimension. Note that because this project has a third variable (not shown), there are repeating events for some visits

more than one lag variable. This means that that lagged and unlagged variable on the grid dimensions do not have to be variations of the same state variable. To make this process easier, the GridWare File Converter described in Chap. 4 can create new state variables from existing GridWare projects. Simply load in all of the trajectory files, click Modify State Variables, and select the "Create Lagged Variable" option. The GWF file that is created is essentially blank, so I recommend just copying the contents of the original GWF file and adding the declarations for the lagged variables (only the names have changed).

As an example, I created three new lagged variables in the Example 1 project we have been using: Variable1LAG1, Variable2LAG1, and Variable3LAG1. Figure 5.2 shows two state space grids using Variable 1, Variable1LAG1, and Variable2LAG1 for the A1_1.trj trajectory. The first grid is a lagged phase plot as described in Chap. 4, with the original raw state variable Variable 1 on the x-dimension and its lag 1 counterpart on the y-dimension. Each node now represents a transition rather than a simultaneous state. So, just observing that the trajectory ends in the 2/4 cell tells you that the last transition was from category 2 to category 4, and this specific transition only occurred once in the sequence. A second key feature to note with these lagged grids is that in duration-based projects the size of the node denotes the duration of the *antecedent*. Finally, because the nodes (events) now depict transitions, the transition lines actually extend to a second-order or lag 2 transition as well. For example, the transition from cell 2/1 to cell 1/2 in the left grid in Fig. 5.2 can be interpreted as a three state sequence for Variable 1: 2, 1, 2. Combining this with the Transitional Measures available in version 1.15a—identifying origin and destination cells—enables the quantification of the frequency of 3-state events. This can be expanded further to 4 or more states as described in Chap. 7.

The grid on the right in Fig. 5.2.should be compared with the grid shown in Fig. 3.6. These are the same data except that Variable 2 is at lag 1 in Fig. 5.2. To make this example less abstract, let us say Variable 1 is the parent behavior and Variable 2 is the child behavior and the values of 1–4 correspond to hostile, withdrawn, neutral, and positive behaviors, respectively. With that in mind, the single event in cell 2/3 means that the parent was withdrawn and then the child was neutral. If we follow the next transition, we now see that the parent's next behavior was hostile. Using GridWare 1.15a, we could compute a Transitional Propensity that reflected the probability of the parent becoming hostile given the parent withdrawn to child neutral transition. The possibilities enabled by lagged state variables are vast, especially when combined with the collapsing dimensions technique described in Chap. 7.

Single-Dimension Analysis

Sometimes you will want to obtain measures based only on one of the dimensions of the state space grid but not the other. For example, with the dyadic grids used for parent–child and peer analyses, you may want a measure of just the parent's measures or just the child's measures. The way to do this is to make both dimensions of the state space grid display the same state variable. Using the axis selection menus (x-axis or y-axis), select the same variable for both dimensions. Now the trajectory or trajectories displayed only visit the cells along the diagonal—it is an $n \times 1$ state space grid. It is impossible for nodes to occur anywhere else. Now you have isolated the one state variable (e.g., child or mother) and can export measures based on this behavior. However, I strongly recommend that you select all of the cells of the diagonal and export the cell and region measures, even if you are interested in the whole-grid variability measures. You can still get all of the same measures for the diagonal region as you can for the whole-grid, but these will be based on the actual number of cells in the $n \times 1$ state space at hand. This can have an impact on any of the measures that use division in their calculation, including Dispersion. Also, keep in mind that the number of events will be affected by the dimension or dimensions not shown, so events will not reflect only the one state variable but all of the variables in the system (i.e., in the trajectory file). Visits, however, is a great measure when using this uni-dimensional approach.

I must emphasize a word of caution with this approach. Sometimes, measures derived from these dimensions are useful (or requested by a reviewer) yet they do not adequately represent the system as you have defined it in state space. The relation between the individual dimensions and the two-dimensional state space is not additive. To quote the Gestalt mantra, "the whole is *different* than the sum of the parts."

Summary Grids

In Chap. 2, I described a variation on state space grids that exploited the features of GridWare to create summary grids (e.g., Lunkenheimer et al. 2011). These are visual matrices that have one plot point per cell centered in that cell which connote some value of that cell for the entire sample. Let us use Example 1 from the GridWare download to see how it this can be accomplished. Open Example 1 in GridWare 1.1b and export measures. In the Export Measures Export Option window, select only All Trajectories as Group, Individual Cell Measures, and Visits, leaving all other check boxes blank. This will save an export file with one case—the average number of visits in each cell for the entire sample. We are going to create a new project.

Open the new exported file with the cell visits in Excel. The first row is the header row and the second row has all the values. That is it. If you do not see that go back and export again. Note that after the few filter variables is a sequence of variables, x1y1Visits, x1y2Visits, etc. You should see the variable naming pattern—*y* changes with each subsequent variable but the *x* value only changes after all the *y* values in that category have been accounted for. That is *x* changes slowly and *y* changes quickly. This is the standard format for all cell identifications. This is also a good reminder that you need to make note of which state variable was presented on the *x*-axis and which was presented on the *y*-axis at the time of export. Otherwise you will get confused about which cells you are analyzing.

The first thing we need to do is to make a cumulative series from those Visits values. In cell F3 type 0 and hit enter. Click in cell G3, just below the label x1y2Visits (G1) and 0.222 (G2), and type the following: = f3 + f2 and hit enter. The value in G3 should now be 0.333. Click on cell G3 again and then hover your cursor over the square in the bottom right corner of that cell until it turns into a crosshair (+). Click and drag that crosshair to the right and stop when the last cell highlighted by the gray box is cell V3, one column past the last value, x4y4Visits. When you release the mouse button, you will see that Excel has updated the sequence in row 3 to a cumulative series ending with the value of 5.665 in cell V3.

The next step is to put these new values into a column. Highlight all of the cells from F3 through V3 with your mouse and copy them (Ctrl-C, right click and select Copy, or use Edit menu). Open a new worksheet in Excel, make cell A2 the active cell, and then select Paste Special from the menu at the top. In the Paste Special window, select Values and check the Transpose box at the bottom. Click OK and now you have a column of values in Column A. Now in cell A1 type "Onset", in cell B1 type "Variable 1", and in cell C1 type "Variable 2." Now, type in the state space grid cell numbers in columns B and C that correspond to those values: B2 = 1, C2 = 1, B3 = 1, C3 = 2, B4 = 1, C4 = 3, etc. You should see the pattern because it is the same as the *xy* names in the sequence that they were exported. Leave cells B18 and C18 blank. Now save the worksheet as a tab-delimited text file: Example1summary.trj (see Fig. 5.2). Save this file in a folder named "Example 1 Visits_trjs".

Now we can make the GridWare project by editing the Example 1.gwf file. Open this file in your text editor and save it immediately as "Example 1 Visits.gwf" in

Onset	Variable 1	Variable 2
0	1	1
0.333	1	2
0.555	1	3
1.333	1	4
1.555	2	1
2.111	2	2
2.555	2	3
3.444	2	4
3.555	3	1
3.666	3	2
4.11	3	3
4.554	3	4
4.665	4	1
4.776	4	2
4.887	4	3
5.443	4	4
5.665		

```
<GridWare>
<Config>
identifier          categorical      Sample
state     integer   variable 1       1      4      1      2      3      4
state     integer   variable 2       1      4      1      2      3      4
MinReturns          2
MaxReturnTime       10
MaxReturnVisits     6
MinEventDuration         0
MinCellDuration     0
MissingValueSymbol       --
</Config>
<Trajectories>
Sample  Filename
Example 1           Example1summary.trj
</Trajectories>
</GridWare>
```

Fig. 5.3 Trajectory file and GridWare file for grid summary of Example 1 Visits

the same directory where you saved the Example 1 Visits_trjs folder. Now edit this file so it looks similar to the gwf file in Fig. 5.3. Now open up our new project in GridWare!

When you click on Show Nodes, you should immediately see that although the size of the node corresponds proportionally to the number of visits in that cell, the nodes are not centered. You could click each cell one by one and rerandomized until it looked pretty much in the center. Alternatively, and more efficiently, you can save the project under a different name (use Save As from the File menu): Example 1 Visits Centered. This will make a new folder in the working directory named “Example 1 Visits Centered_trjs.” Open that folder and then open the file Example1summary.trj in your text editor. Three new columns have been added: color, Variable 1 POS, and Variable 2 POS (POS = position). The color column is the RGB number corresponding to the default blue of the nodes. The POS columns have the relative position along the cell axes ranging from 0 (x left or y bottom) to 1 (x right or y top). Change all of these values in both columns to 0.5. That will center every node in every cell. When you reopen the project in GridWare, Show Nodes, and select Unmark Start Node then your grid should look similar to the one in Fig. 5.4

In this example, we see that there were not many visits in the bottom right quadrant but there were many visits in all of the cells in category 3 of Variable 2. The trick with summary grids is to make the onset values equal to the progressive cumulative sum of the values you want to display. These can be any value on any scale. You could make summary displays from values not even related to state space grid variables (e.g., population values on a 2×50 grid of states and gender).

The Special Case of Ordinal Time Series

When state variable categories are truly ordinal—an interval sequence of integers which correspond to units of constant size—there is at least one more way to analyze the state space grid data. In this case, by ordinal I mean truly ordinal by

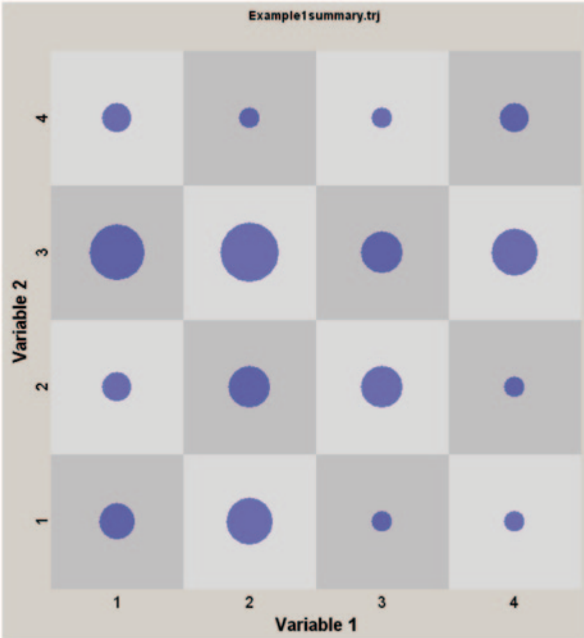

Fig. 5.4 Example 1 summary of visits

measurement. In the Lewis et al. (1999) study, for example, the dimensions of the grid were ordinal categories of distress and attention. But while these categories were sequential in terms of relative differences to lower and higher values (e.g., 1>2>3>4>5), the units themselves were somewhat arbitrary. They were applied by coders as a way of approximating ordinality. The *magnitude* of change between 1 and 2 or 4 and 5 was not constant because the units were not measurement quantities. Thus, these ordinal variables are not the special case.

What is special is when the state variable categories reflect a measurement quantity, like a count or a test score. For example, suppose that the axes of a grid were a state anxiety score and the grade on a series of quizzes administered once a week for an academic year. Each student's trajectory on the grid would reflect the simultaneous variability in anxiety on the day of the quiz and the quiz score. With these state variables the difference from one category to the next would be a unit, a quantifiable magnitude reflecting some real world measurement. The change from 60 to 61 would be equivalent to a change from 90 to 91. That is, now *distance is meaningful*. Each *xy* coordinate is exactly that, a coordinate, and can now be treated in the same way as physical coordinates in space. We can compute each transition as a distance in the same way as we compute the hypotenuse of a right triangle: the square root of ($\Delta x^2 + \Delta y^2$). That is, the length of each transition line is a computable value and the total distance that the trajectory travels through state space can be derived from the sum of these transitional distances: the greater the distance covered, the more variable the behavior of the system. While this computation was not provided in GridWare, it is easily computed in Excel on each trajectory file event-by-event. If columns B and C were our state variables of anxiety and quiz score, for example, the formula in column D (starting at row 3) would be: =SQRT(((B3−B2)^2)+((C3−C2)^2)). Dragging that formula down the rest of the rows in column D would create the distance of each event-to-event transition. The sum of that column D would be the total distance score, an index of total variability.

Filter or State Variable?

In some projects, it is possible to have variables that can function either as state variables or filter variables. For example, suppose you have 30 min of observational data on parents and children as they proceed through a series of 10-min activities: free play, discussion, and playing a game. Each dyad could have one long trajectory file in which activity type is a category in the Activity state variable—a third variable after the Parent-Affect state variable and the Child-Affect state variable. The events in the first 10 min would all have the same value for Activity (play), the events in the second 10 min would have all the same value (talk), and the events in the final 10 min would all have the same value for Activity (game). With Activity as a state variable, it would be possible to analyze Child-Activity states or Parent-Activity states by selecting them for display on the *x* and *y* dimensions. The dyadic Parent–child grid could also be color coded by activity to display how affected patterns differed by activity. However, to derive dyadic measures with activity as well is not possible—if parent and child are the dimensions, then activity is not involved in any quantification.

One solution would be to create 3-dimensional state space (see Chap. 7). Another would be to export measures using the time window—each 10-min segment separately. This could work, but often the length of activities in laboratory research varies by even a few seconds making this a less optimal solution. A better solution in this case is to save each trajectory as three separate trajectories, one per activity. Now Activity is a filter variable that uniquely identifies each trajectory and is included in the trajectories list in the GridWare File. The visual display will be virtually the same as before (except for the start node, if marked) when selecting across all three activities. However, when exporting, you will get a different value for each activity (each dyad-activity combination is a case).

For most state space grid projects, I usually have a few versions that vary in terms of the number of filter variables and/or state variables (e.g., lags). Especially, when first starting a new project, I find it helpful to "see" the data from different angles. In general, though, I find that projects with the shortest trajectories are more useful than those in which the data are in one long trajectory. This of course depends on the nature of the data and the research questions. It is up to you to work out what is the best approach.

Chapter 6
Between-Grid Analyses

In this chapter, the basic analyses described in Chap. 5 are extended into comparisons across grids. These mostly person-centered analyses can be comparisons across different tasks or across measurement occasions as in a longitudinal study. The nature of these analyses often requires transforming the default stacked data format from the Gridware export into an unstacked format (see Box 6.1) and/or the exportation of individual cell measures (available only in version 1.1b, see Box 3.3). The fundamental question underlying these kinds of analyses can be stated as "how are the trajectories on two grids different from one another?" In this chapter, I will cover three ways that this has been done: profile analysis, intergrid distance scores, and cluster analysis.

Box 6.1 Stacked and Unstacked Data GridWare exports measure in a data matrix based on the trajectories list. In most cases, this results in a stacked file in which each row (case) corresponds to a trajectory, with multiple rows for each primary case identifier, for example, with parent-child dyadic interaction data, there is often a set of two or more interaction tasks. With these tasks as filter variables, the primary case identifier, dyad ID, repeats on multiple rows, one for each interaction task. Thus, this is called "stacked" (or long or person-period) data.

For statistical analyses, most often we want unstacked (or wide or person-level) data where the different tasks are denoted by different variables and there is only one row of data for each primary case identifier. Fortunately, it has gotten a lot easier to switch back and forth between these data structures in SPSS. Using the data restructure wizard, you can take the stacked data file from GridWare and make it unstacked in SPSS. Here are a few tips for working with stacked GridWare exported data. First, you want to go from cases to variables, in SPSS parlance. Next, the identifier variable is the one (or ones) that you want to keep when there is only one row per case. Almost always this is some sort of ID variable (e.g., Fam1234). You can also include variables that are constant for that ID (e.g., gender). The index variables are those filter variables for which you want a separate variable per measure. So,

T. Hollenstein, *State Space Grids*, DOI 10.1007/978-1-4614-5007-8_6,

if in the stacked version you had three discussions (filter), one per row/case, for an export of the visits variable, in the unstacked version there would be one row/case with three visits variables, each corresponding to a discussion (e.g., visits.1, visits.2, visits.3). There are several other options to choose as you progress through the wizard but mostly the defaults work fine enough.

Profile Analysis

The most basic way to analyze changes between state space grids is to subject the measures derived from each to standard statistical comparisons. These can be measures of structure (i.e., whole-grid measures) or content (i.e., cell or region measures), for example, in the Hollenstein et al. (2004) study, we had 1 hour of coded observations of parent-child interactions on each of two occasions that occurred within the same month. This enabled the calculation of test-retest reliability of the state space grid variability measures (cell range, transitions per minute, and average mean durations) via correlations. These correlations were particularly strong for observational measures, ranging from 0.4 to 0.6 across observation sessions. Thus, whole-grid measures appear to be relatively stable over short periods of time.

It is most common, however, for state space grid research to focus on contextual task-to-task differences in system behavior, or on developmental changes in longitudinal designs (i.e., micro, meso, and macro relations as in Fig. 1.1). In these cases, a repeated measures ANOVA is used for profile analyses to examine the grid-to-grid changes in the same measure. Figure 6.1 shows two examples of my own work, with tasks to test quadratic predictions of change in dyadic variability. Using an A-B-A design as described in Chap. 2, where B is a task designed to elicit negative emotions (e.g., conflict), the change in transitions and dispersion in the hypothesized quadratic high-low-high form was the opposite shape of the negative emotion profiles (Hollenstein and Lewis 2006). Thus, the quadratic F-test indicated main effects for the variability measures and for the profile interaction with stress (top row of Fig. 6.1). These mother-daughter dyads participated in the same discussion sequence three more times, every 6 months (Hollenstein 2007). Because the girls all changed schools at wave 2 (grade 7), the hypothesized developmental phase transition was expected to occur during that year. As shown in Fig. 6.1 (bottom row), all dyads increased in negative affect but only the dyads with relatively low stress girls showed the hypothesized quadratic profile. Taken together, these two studies demonstrate the utility of profile analysis for testing DS hypotheses of change and stability at the meso and macro levels.

Another method to analyze grid-to-grid changes would be through multilevel modeling for repeated measures. With this approach, level 1 variables would be the measures obtained for each measurement occasion (e.g., task, longitudinal wave). Level 2 would correspond to between-subjects factors. The nested nature of mul-

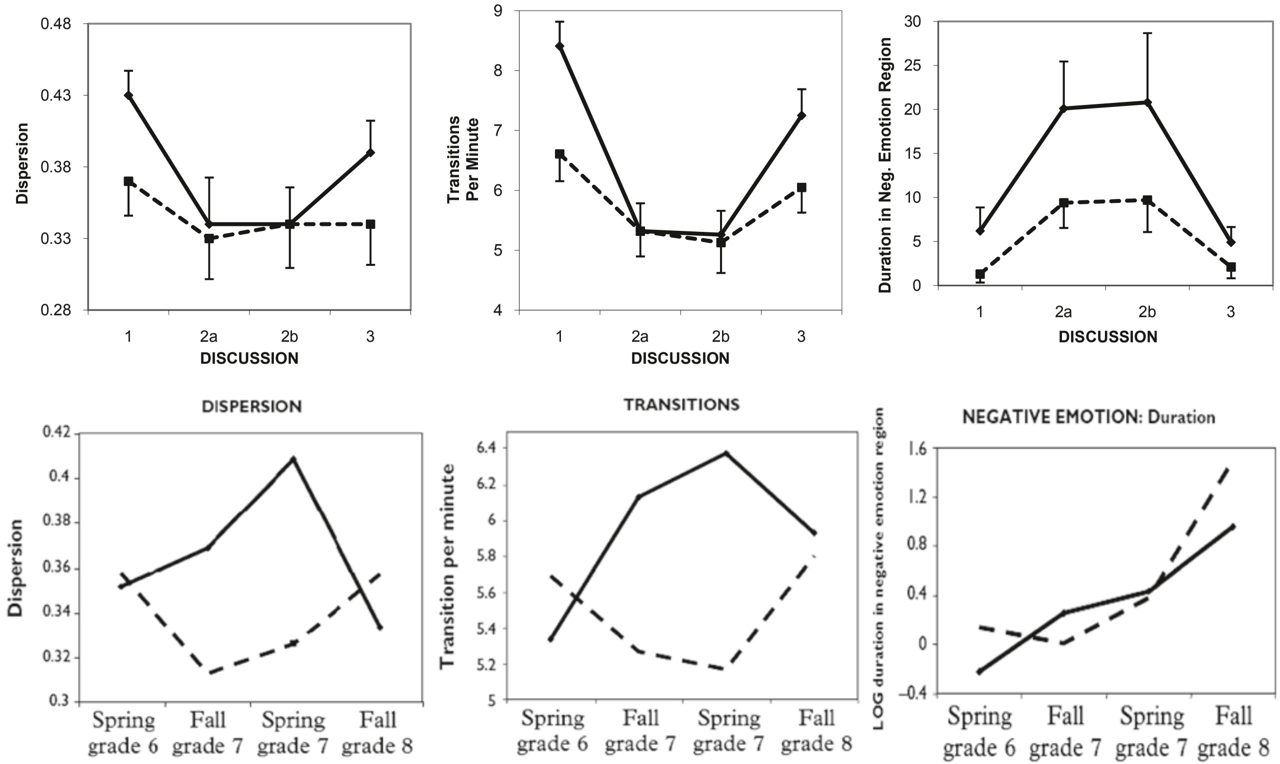

Fig. 6.1 Mother-daughter dispersion, transitions, and negative emotion across three discussions (positive, first half of conflict, second half of conflict, positive) at wave 1 (*top row*) and across four longitudinal waves of the conflict discussions (*bottom row*). Solid lines represent dyads with girls in the high stress group and the dashed lines represent dyads with girls in the low stress group (< 2 stressful events). In the longitudinal study stressful events were those at wave 2, coinciding with a school transition

tilevel modeling also makes it correspond well to the model depicted in Fig. 1.1: microscale indices of variability can change across mesoscale contexts at level 1 and these are all nested within the longitudinal macroscale repetitions at level 2. However, while constraining the parameters to nonlinear relations such as a quadratic is possible with multilevel modeling, there would still be the need to interpret these effects with the aid of visualization of the profiles. Thus, a combination of procedures—as is common in multivariate analyses—is often preferable.

Profile and other statistical approaches to comparing trajectory behavior across grids function at the aggregate level. They necessarily look for significant group effects. However, most DS research questions are focused on processes that occur at the individual level, to capture the idiosyncratic behavior of each system, for example, trajectories on two grids could produce identical values for a whole-grid measure (e.g., transitions) yet cover the state space in very different regions. Thus, to make grid-to-grid comparisons at the individual level, we must approach the problem differently. Next, I will describe two such solutions, intergrid distance scores and cluster analyses, developed to quantify qualitative differences across state space grids.

Intergrid Distance and Cluster Analyses

The intergrid distance score (IDS) and cluster analysis techniques were created to detect qualitative differences in trajectory patterns between grids. The attempt here is to combine the information revealed by the trajectory on the grid in a way that retains both content and structure at the same time with one quantitative (IDS) or qualitative (cluster) index. As described in Chap. 2, these techniques were developed for the Lewis et al. (2004) study to examine month-to-month change in infant socioemotional behavior. These techniques can be used for any comparisons between two or more grids of the same order (i.e., both grids must have the same number of categories on the x-dimension and the same number of categories on the y-dimension) with trajectories of the same duration. The nature of the grids being compared depends entirely on the study design and research questions. Comparisons could be within or across individuals or dyads, task-to-task differences or the same task repeated two or more times, or they could even be used to compare subsets of trajectories (e.g., first half versus the second half of a parent-child interaction).

Intergrid Distance

IDS is based on a simple Euclidian distance premise (much like the total distance score for the special case of ordinal variables discussed in Chap. 5) to measure the net difference between two values. In this case, we have as many values as we have cells (as few as four with the most basic 2×2 grid). The values in the cells can be any one of the values exported for individual cells in version 1.1, though my preference is for duration per visit as this combines both duration and frequency information into one mean duration value. Keep in mind that many cell variables can come up as missing values if there is division by zero (e.g., when visits = 0 for the duration per visit measure) or there is a criteria for calculation (e.g., minimum number of returns for return time). Recoding the missing to a value of zero is the solution in whatever program you use for computing IDS.

Figure 2.6 illustrates the general pattern of IDS comparisons. Each cell is subtracted from the corresponding cell in the other grid. It might seem appealing that this simple subtraction of values retains directionality. Positive and negative values would reveal whether the values were higher or lower in one grid or the other. This is fine at the individual cell level, but if we are going to sum these values up to one index, then they would necessarily sum to zero (assuming both trajectories were of the same length or duration). To get around this directionality problem, the differences can be squared to obtain positive values of zero or greater for every cell comparison. The squared differences for every cell can then be summed up into one IDS value:

$$IDS = \sum_{i=1}^{\#\ of\ Cells} (A_i - B_i)^2,$$

where A_i is the value in cell i from one state space grid and B_i is the value in the corresponding cell i from the other state space grid. Thus, two identical trajectories would have an IDS value of zero and positive values reflect the degree to which two trajectories differ from one another because these values are dependent on the scale of measure used, the absolute magnitude of this value is not meaningful (much in the same way that statistical variance values are not meaningful as absolute values). The relative value in the context of other equivalent comparisons does have meaning, for example, as Lewis et al. (2004) showed, the highest month-to-month IDS values occurred during a developmental phase transition when behavior was expected to be more variable and less predictable.

Uneven trajectory lengths are a problem because the durations will necessarily be different between the two grids, for example, for grids based on tasks that are 4 min and 8 min long, there is twice as much time in the latter. Cell durations are necessarily going to be higher in that second grid and an IDS score would not account for that. As a solution, a variation on the IDS procedure is to first re-express the cell values as within-grid standardized scores. Instead of using the raw durations or visits in each cell for the IDS calculations, standardizing would rescale the measures as standard deviations from the mean of that trajectory. The result within each grid is a set of values that are both positive and negative. Even with negative cell values, IDS can then be calculated as described above.

Cluster Analyses

Cluster analyses have long been used to identify qualitative differences in a pattern of variable values. There are many variations but they all operate with the same underlying premise: minimize the differences within a cluster and maximize the differences between clusters . Because it is often desirable to discriminate trajectory patterns on state space grids, cluster analyses are useful tools for grouping sets of grids together as homogenous sets. In the old days, when first developing the state space grid technique, I would print out each grid and then visually try to match them together. Then, I looked for quantities that would differentiate these groups. Some of the whole-grid variables were developed this way. Of course, this is a cumbersome process of a consummate state space grid geek and I do not recommend it. Instead, it is preferable to use one of the many clustering options out there.

The product of a cluster analysis is the identification of each case as a member of one and only one cluster. For state space grid analyses, cases are individual trajectories. So, for example, in a sample using a 4×4 grid, there would be 16 values for each case corresponding to each trajectory's value for a given measure (e.g., duration per visit). Note that, as with IDS, there are the same considerations about which individual cell measures to use. The cluster analysis would be run on these 16 variables and each case would be assigned membership in one cluster. Trajectories within a cluster should look quite similar and trajectories that do not share the same cluster membership should look quite different from one another.

The big question is how many clusters in total are best. This is not an easy question to answer and is a major limitation of cluster analyses. Until recently, most cluster solutions combined multiple analyses, reasonable but arbitrary criteria, and convergent validity. Recent advances in covariance matrix analyses (e.g., SEM) have opened up more empirically driven ways to decide on how many clusters fit best. In order to describe the cluster change score used in the Lewis et al. (2004) study, I will detail the "old school" clustering method employed for those analyses. At the end of this section, I will describe the newer covariance methods as a suggestion for how these analyses may be conducted in the future.

Clustering techniques require that you prespecify how many clusters should be extracted. I typically repeat these analyses several times, starting with a 2-cluster solution and progressing up one at a time to six or more clusters per analysis. Each case (trajectory) gets a cluster number to indicate membership. The first thing to look for is the distribution across clusters. I prefer solutions with the highest number of clusters that have a sufficient number of cases per cluster. For statistical analyses, it is necessary to have at least five, preferably ten, cases per cluster. As the number of clusters increases, the more likely it is that one or more clusters will have fewer than five cases. It is possible that the cluster with only one member is an anomalous case, reflecting a significant deviance from the rest of the sample due to measurement error or uncontrolled experimental conditions. It is also possible that this case is a real instance in the population but one that does not occur often. From a person-centered perspective, this case must remain distinct rather than grouped with other cases or dropped as a missing case. This is a difficult and uncertain choice. I recommend consulting some of the fantastic work done by Magnusson and colleagues in thinking about categorizing individuals (Bergman et al. 2003; Magnusson 1999, 2003). The basic choice between the representativeness of odd cases and the pragmatism of making inferences from statistical analyses is up to the analyst.

For state space grid analyses, there is the added benefit of visual examination of cluster groups. While this can help to resolve issues of clusters with a low number of cases, it is primarily a way to name or identify clusters for later analyses. Put these cluster membership categories in as filter variables in the trajectories list in the GridWare file and then explore in GridWare. This approach allows you to converge on an optimal solution of distinct and homogenous groups based on the pattern of behavior in state space. The final verification of cluster analyses, however, comes with convergent validity. Given the patterns that typify each cluster, group differences should conform to well-established measures that would be expected to vary by those groupings.

In the past decade, latent profile analysis (LPA) has emerged as a useful alternative to traditional cluster analyses (Muthén 2001). With LPA, the same goal—minimal within-group differences and maximal between group differences—is executed using maximum likelihood algorithms. However, there are several means for establishing the optimal number of classes (clusters). Iterations begin with a one-class model, and subsequent classes are added one at a time until the addition of a class does not improve model fit. Model fit is assessed by the Bayesian information criterion (BIC; Schwarz 1978), adjusted Bayesian information criterion (adjusted BIC; Sclove 1987), and Akaike information criterion (AIC; Akaike 1973). An increase of one class is considered a better fit if these information criteria are lower than the previous model with one fewer classes (Nylund et al. 2007). Entropy (similar to but not the same as the whole-grid measure) indicates how well a model classifies individuals into groups, with better classification indicated as values approach 1 (Celeux and Soromenho 1996). Two significance tests can be used to compare models. Significant p-values on the Vuong–Lo–Mendell–Rubin (VLMR) likelihood ratio test and the adjusted Lo–Mendell–Rubin (adjusted LMR) likelihood ratio test indicate that the estimated model provides a better fit to the data than a model with one fewer group (Nylund et al. 2007). Thus, with multiple indices, there is now a more standardized and reliable method for determining the best-fitting number of clusters or classes.

For this chapter, however, we are not concerned with stopping at the point of identifying clusters. We are interested in using cluster or class membership as a way of indicating grid-to-grid change. To create a *cluster change score*, you must put *all* of your trajectories into one overall cluster analysis. I emphasize *all* because it cannot be done by running the analyses separately by task or longitudinal wave. The best approach, provided you have all of your trajectories in one project, is to export the individual cell values from GridWare and to keep them in the stacked format (see Box 6.1) for computing the cluster membership. Once you have completed the range of cluster solutions, you can then unstack the data for the between grid analyses. In the end, you should have one variable column per cluster solution for each task/wave. For example, if you converged on a 4-cluster solution and you have three longitudinal waves, then at this point you should have three variables: wave 1 cluster, wave 2 cluster, and wave 3 cluster. There are four possible values for each of these variables corresponding to the membership for each case. Now you are ready to compute cluster change.

For each case, if two adjacent trajectories (e.g., wave 2 and wave 3) are members of the same cluster, then the cluster change score is 0. If, however, the two trajectories are in different clusters, then the cluster change score is 1. Do this for each two-variable combination. In the previous example with three longitudinal waves, this would mean there would be two cluster change scores: one for wave 1 to wave 2 change, and another for wave 2 to wave 3 change. In the Lewis et al. (2004) study, we had one trajectory per month per child from 14 to 24 months. Thus we computed change from 14 to 15 months, 15 to 16 months, and so on. The average change across trajectories, then, is a value between 0 (all trajectories remain in the same cluster) and 1 (all of the trajectories change clusters). Depending on the research question, changing clusters could be seen as an adaptive response to changing environmental demands or as a lack of stability in circumstances which should elicit similar patterns of responding.

Conclusions

The analysis of grid-to-grid change is rather open ended. It is limited only by the research question and the nature of the measurements involved. Here I have described a few approaches to the types of questions that developmental scientists tend to ask. The utility of these approaches is, of course, not limited to developmental analyses but any research for which multiple trajectories per case exist.

At this point, this book could conclude and you would have all the information you would need to conduct state space grid analyses. You have learned how to design measurements, format data, import and export data using GridWare, and conduct basic state space grid analyses. There are several important extensions of this technique that are worth considering, however, and I will cover these in the next chapter on advanced analysis.

Chapter 7
Advanced Analyses

In this chapter, I will describe more advanced techniques for state space grid analysis that go beyond what we have covered in Chaps. 5 and 6. These techniques are grouped in three main areas. First, there are many applications for which the limitation of a two-dimensional state space is a critical problem. The good news is that there are ways to work around this limitation to conduct analyses on three or more dimensions. Next, the problem of continuous, rather than categorical, state variables is discussed with suggestions of how to work with these data. Finally, I will briefly review a set of analytical techniques that are quite complementary to state space grids. These are not exhaustive lists of possibilities, but they cover the range of extensions of state space grid analyses. As the second decade of state space grid analysis begins, I expect that these are the avenues that will show the most growth.

Beyond Two Dimensions

So far, we have only considered the two-dimensional state space. This is partly due to the fact that this is the most basic dimensionality with which to understand the concepts and analyses of state space grids. It is also partly due to the limitations of GridWare and the difficulty of visualizing (and conceptualizing) trajectories in three or more dimensions. Fret not. There are ways to solve the problems that arise beyond a two-dimensional state space ranging from the simple to the complex. Let us start with the simple.

Color for the Third Dimension

The most straightforward way to depict a third dimension on your two-dimensional computer screen is through indicators of shape, size, or color. For GridWare, color is the exploratory option. Open the Example 1 data set that comes with the download of GridWare. You have probably noticed by now that this ex-

T. Hollenstein, *State Space Grids,* DOI 10.1007/978-1-4614-5007-8_7,

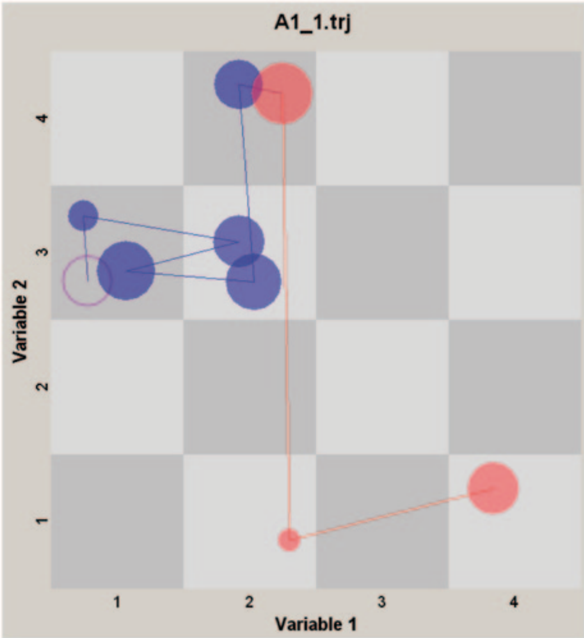

Fig. 7.1 Using color to analyze third dimension. Variable 3 (not displayed on an axis) has been colored to represent the four categories (*1* red; *2* purple; *3* blue; *4* green)

ample project has a third state variable, Variable 3, which does not appear when you first open it in GridWare. To view this state variable, use one of the axis selection menus (*x*-axis or *y*-axis) and select Variable 3. Let us suppose that Variable 1 represents the child's behavior, Variable 2 represents the Mother's behavior, and Variable 3 represents the Father's behavior. Hence, these data might correspond to a triadic family interaction. Let us further suppose that categories 1 through 4 represent affective valence: hostile, withdrawn, neutral, and positive (remember, you can Alt-click on the axis labels to edit if you like). We cannot view all the three dimensions at once, only two, because of the way this project is formatted. So, if we were interested in how the father's affect corresponded to the mother's and the child's, then we can color the father's categories and display the two-dimensional mother–child grid. While Variable 3 is displayed on the state space grid, color category 1 red, category 2 purple, and category 4 green (leave category 3 as blue). Now, return to the display so that Variable 1 (child) is on the *x*-axis and Variable 2 (mother) is on the *y*-axis. Et voila! Three-dimensional! Selecting only trajectory A1_1 that we have seen before (Fig. 3.6), it should look similar to Fig. 7.1. Continuing with this hypothetical example, the mother (Variable 2) gets mutually hostile with the father (red) but the child (Variable 1) is simultaneously positive. We can also see that, because there are no green nodes, the father did not exhibit positive behavior.

The use of color in GridWare is purely visual. It facilitates exploratory analyses and is helpful for creating images for presentations or manuscripts. However, the selection of color has no impact on any measures and so has important limita-

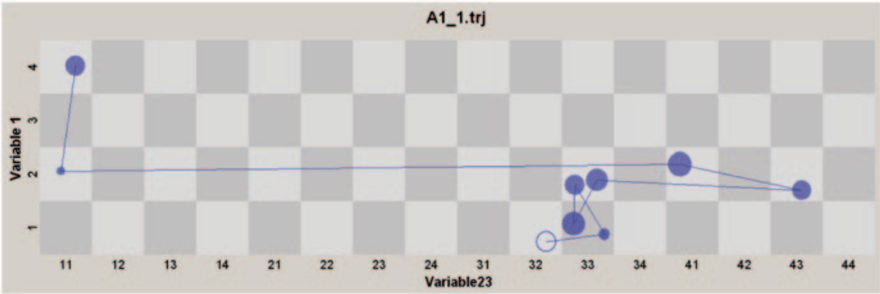

Fig. 7.2 Three dimensional state space grid for Example 1 data. Variable 2 and Variable 3 were concatenated to form state categories of a new Variable 23. The first digit in each *x*-axis category label corresponds to the value of Variable 2 and the second digit corresponds to the value of Variable 3

tions on the three-dimensional analyses. Fortunately, we have other tools at our disposal.

Collapsing Dimensions

The difference between a state that is a category of a state variable and a state that is a cell on a state space grid is a levels-of-analysis issue. Both are "states" but one is higher order and another is lower order. So, it is with a nested, hierarchical structure. When confronting the two-dimensional limitation, we can use multiple levels of states to our advantage. If the cells are states, then they could also be states of a state variable on a state space grid dimension. Let us walk through an example.

Open Example 1 again and display Variable 2 on the *x*-axis and Variable 3 on the *y*-axis. There are 16 cells and we can identify these by the traditional *x*/*y* naming convention. So the cell in the top left would be identified as cell 1/4 (or 14 for short) because it is a combination of state category 1 for Variable 2 and state category 4 for Variable 3. In this way, each cell can be identified by the *x*/*y* combination. That is each cell can be represented by the *concatenation* or appending of the *y*-axis category after the *x*-axis category. This is a simple process using Excel or the GridWare File Converter (see Chap. 3). With a new 16-category state variable which is the combination of Variable 2 and Variable 3, Variable 23, we can now view our Example 1 data in three-dimensions (Fig. 7.2).

This remarkable advancement is really not so astounding once you break it down. As I mentioned in Chap. 2, the unit of analysis is always the cell. The arrangement of the cells, even if based on ordinal state variables, is arbitrary for calculations (except for distance, as described in Chap. 5). Visually, a state space grid with random arranged axis categories would be difficult to interpret. Computationally, because adjacency of cells has no mathematical meaning, the cells can be arranged any way at all. Perhaps the best way to understand this is through another example. In our most recent study, Lavictoire et al. (2012), we examined triadic peer interactions of young children. For analytic purposes, there was one child deemed the target child and two peers, Peer 1 and Peer 2, and all were coded for affect with SPAFF (Gottman et al. 1996). State space grids were created for each dimension (Target, Peer 1, and Peer 2) with four categories: aversive (AGGR), withdrawn (WITH), neutral (NEU), and positive (POS). Thus, we had a $4 \times 4 \times$ *cube*. Figure 7.3 shows how this cube of 64 cells can be conceptualized as a combination of three two-dimensional grids derived from the *x*, *y*, and *z* planes of the cube. In this example, the blue side represents the space formed by the intersection of the Target and Peer 1 axes (collapsing across Peer 2), the red side represents the space formed by the intersection of the Target and Peer 2 (collapsing across Peer 1), and the yellow side represents the space formed by the intersection of Peer 1 and Peer 2 (collapsing across the Target). The bottom state space grid in Fig. 7.3 shows the result of concatenating the Peer 1 and Peer 2 affect categories to make an *x*-dimension of 16 categories. The trajectory displayed on all four of these grids is the same, just from different angles including different sets of cells.

Unlike the mother–father–child instance in Example 1 described above, the Lavictoire et al. (2012) study was not concerned with the differentiation of Peer 1 and Peer 2. Thus, reciprocal states (e.g., withdrawn/positive and positive/withdrawn) were functionally redundant. For this study, we were able to collapse these states as functionally equivalent in reference to the target and reduce the 4×16 state space grid to a 4×10 grid. This was a special case due to the nature of the extant data. Regardless of these circumstances, however, analysis of three-dimensional grids is pretty much the same as any other state space grid analysis previously described. In the peer study, for example, we examined both whole-grid variability and specific content states related to the target child's externalizing and internalizing problems.

Multidimensional Lag Transition Grids As you come to understand the implications of representing more than two dimensions on state space grids, it may occur to you that, by extension, this collapsing dimensions idea could be accomplished multiple times. This is true, though there is likely a ceiling of interpretability after about two or three iterations of this process. Because the number of categories increases somewhat exponentially as you make cells into dimensional categories, it is best to have as few original states as possible. In one step, for example, a 3×3 grid can turn into a 9×9 grid and by the third step an 81×81 grid. In order to understand this process, it is best to work through an example together. To do so, let us start with a brand new project (Box 7.1). This is the last chapter so you should be able to start a new project with flawless expertise (note that no Preferences are declared in the GridWare File so the project will just run with the default values). Once you have opened this one-trajectory project, continue reading.

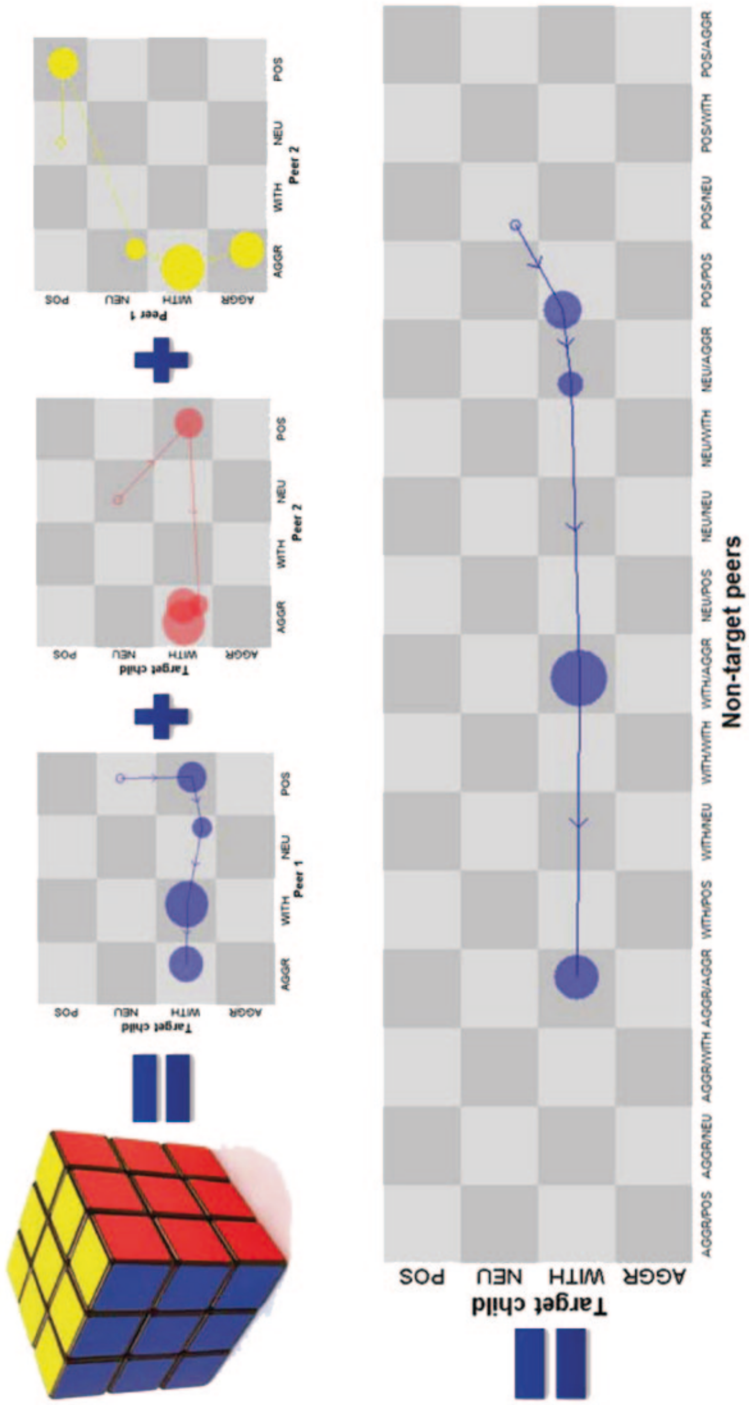

Fig. 7.3 Derivation of a three-dimensional state space grid displayed on two dimensions

BOX 7.1 New Example Project Below are a GridWare File (top) and Trajectory File (bottom) to create a new example project for multidimensional lag transitions and cumulative change state space grids. This example illustrates the range of possibilities for extending state space grid analyses through the use of cells as states and the differences between states as new categories. The data here are based on a simple valence coding of the behavior of two children: negative (−1), neutral (0), and positive (1). Thus, these can be thought of as both nominal and ordinal categories. Note that these state variables have to be declared as categorical (rather than ordinal) due to the use of negative values. All of the state variables to the right of column three are derived from the observed states for Kid1 and Kid2 state variables.

<GridWare>											
<Config>											
Identifier	Categorical	ID	KID12								
State	Categorical	Kid1	−1	0	1						
State	Categorical	Kid2	−1	0	1						
State	Categorical	Kid1lag1	−1	0	1						
State	Categorical	Kid2lag1	−1	0	1						
State	Categorical	Kid1lag2	−1	0	1						
State	Categorical	Kid2lag2	−1	0	1						
State	Categorical	Kid1lag01trans	−1/−1	−1/0	−1/1	0/−1	0/0	0/1	1/−1	1/0	1/1
State	Categorical	Kid2lag01trans	−1/−1	−1/0	−1/1	0/−1	0/0	0/1	1/−1	1/0	1/1
State	Categorical	Kid1cumchange	−3	−2	−1	0	1	2	3		
State	Categorical	Kid2cumchange	−3	−2	−1	0	1	2	3		
State	Categorical	Kid1relchange	−2	−1	0	1	2				
State	Categorical	Kid2relchange	−2	−1	0	1	2				
</Config>											
<Trajectories>											
ID	Filename										
KID12	Multilagexample.trj										
</Trajectories>											
</GridWare>											

Onset	Kid1	Kid2	Kid1lag1	Kid2lag1	Kid1lag2	Kid2lag2	Kid1lag01trans	Kid2lag01trans	Kid1cum-change	Kid2cum-change	Kid1rel-change	Kid2rel-change
0	0	0	−1	0	−1	1	0/−1	0/0	0	0	0	0
1	−1	0	−1	1	−1	−1	−1/−1	0/1	−1	0	−1	0
2	−1	1	−1	−1	0	−1	−1/−1	1/−1	−2	1	0	1
3	−1	−1	0	−1	1	−1	−1/0	−1/−1	−3	0	0	−2
4	0	−1	1	−1	1	0	0/1	−1/−1	−3	−1	1	0
5	1	−1	1	0	0	0	1/1	−1/0	−2	−2	1	0
6	1	0	0	0	0	1	1/0	0/0	−1	−2	0	1
7	0	0	0	1	1	1	0/0	0/1	−1	−2	−1	0
8	0	1	1	1	–	–	0/1	1/1	−1	−1	0	1
9	1	1	–	–	–	–	–	–	0	0	1	0
10												

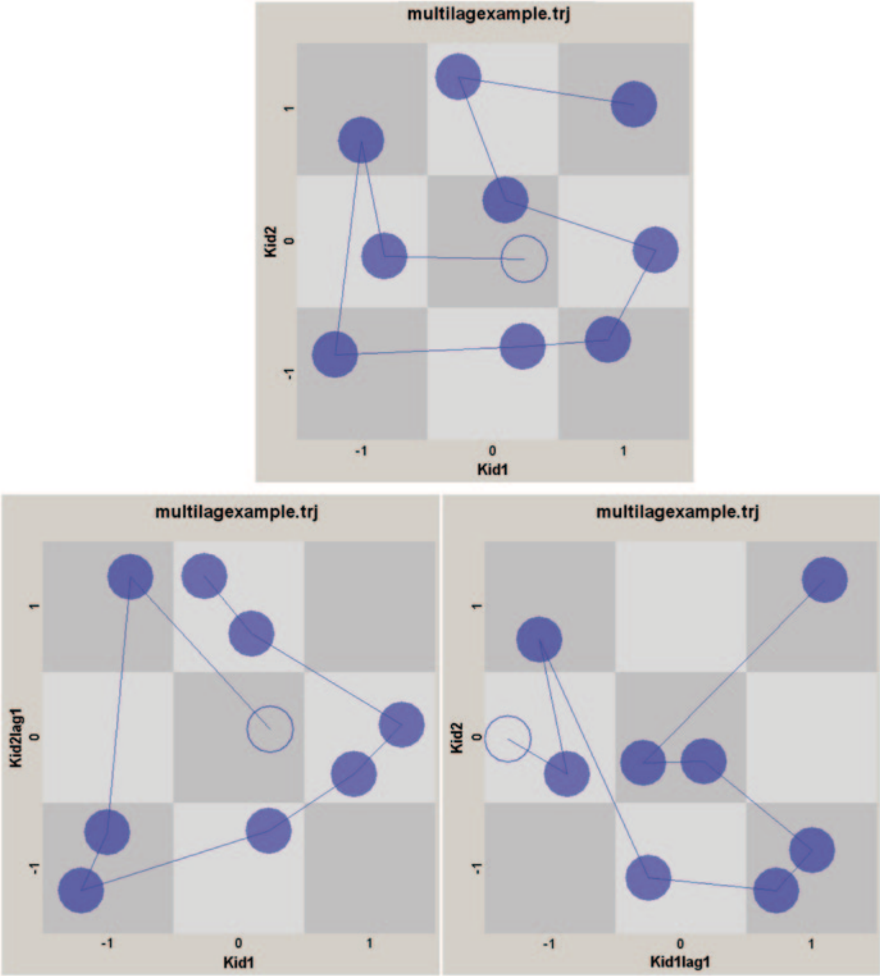

Fig. 7.4 Original and cross-lag state space grids for the multilag example project. The top state space grid displays the original, simultaneous behavior of Kid1 and Kid2. The bottom state space grids display both cross-lagged grids

For this example, I created hypothetical data of two children getting into and out of a fight. Their behavior is coded as valence in numerical terms such that −1 is negative behavior, +1 is positive behavior, and 0 is neutral behavior. Thus, as is displayed in the grid that opens with the project (Fig. 7.4), we see the peer dyad move from mutual neutral to mutual negative to mutual positive in a series of ten steps. Visually, we can trace these steps and see quite clearly in this idealized example how Kid1 starts the fight but is also the first to move toward resolution. Quantitatively, however, we need to extend beyond the simultaneous grid to lagged state space in order to capture these transitions (see Chap. 5 for description of how to lag

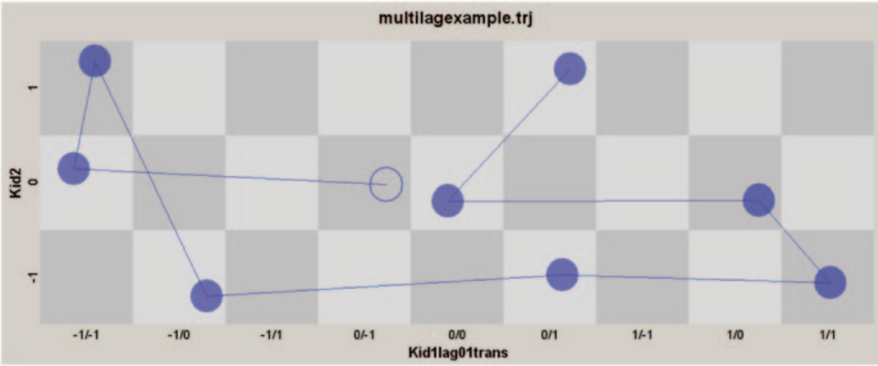

Fig. 7.5 Lag transition grid of lag1 transitions for Kid1 with Kid2 behavior

variables). Thus, as shown in the lower part of Fig. 7.4, we first need to create lag 1 variables for each child: Kid1lag1 and Kid2lag1 (Box 7.1). With these lagged grids, we can start to see the transitional pattern. For example, comparing the left column of the Kid1 with Kid2lag1 grid with the bottom row of the Kid1lag1 with Kid2 grid, we can see that Kid1 never followed Kid2's negative behavior with negativity of his own, but Kid2 did (see cell −1/−1 in both lag grids in Fig. 7.4). Note that we can continue to lag beyond one step, as can be seen by the lag2 variables that come next in this trajectory file.

Now that we have the lagged variables for each child, we can proceed to the next step. Select the axes of the state space grid in GridWare so that the *x*-axis is Kid1 and the *y*-axis is Kid1lag1—a within-variable lagged state space. Here you see the sequence of Kid1's behavioral stream displayed as a lagged phase plot. Each cell denotes a one-step transition. The start node, for example, is in cell 0/−1, reflecting the first transition from neutral to negative behavior. This cell can now be identified as the 0/−1 state. In fact, all nine of the cells now can be identified that way. This is how the two variables in the new project, Kid1lag01Trans and Kid2lag01trans, were created. These are now nominal state variables with nine categories each.

There are (at least) two ways that these data might be analyzed. First, if there is an explicit direction that is of interest such as Kid2 is the older sibling and we are interested in how the younger sibling's behavior follows, we would want to see Kid2's behavior relative to Kid1's transitions (Fig. 7.5). Here we can see that when Kid2, the older sibling, was negative, Kid1 transitioned into less negative states (−1/0 and 0/1) or stayed positive (1/1, see bottom row of Fig. 7.5). With this grid, we can see the antecedents of both children and the behavior consequence of one of them. The second way we might want to analyze these data is to examine both the antecedent and consequence for both children, as shown in Fig. 7.6. This grid contains a lot of information. Through cell and region selection, we could identify any antecedent–consequence combination we wished. These selections might be specific, like the count of instances in which both the children were becoming more negative, or mismatched transitions of increasing positivity with increasing nega-

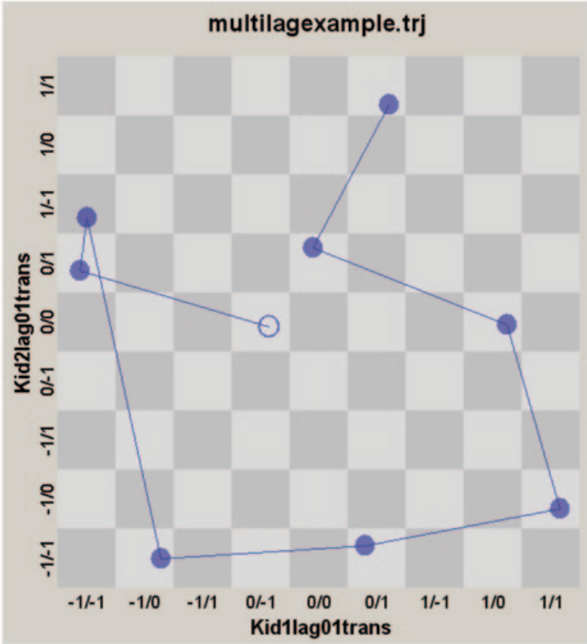

Fig. 7.6 State space grid of lag 1 transitions for both Kid1 and Kid2

tivity. Exported cell frequencies and durations from transition grids similar to the one displayed in Fig. 7.6 would provide all of the necessary information for any lag sequential analyses you wish to conduct.

Of course, this process can continue, with lag 2 and beyond. As you add these lags, you can identify multistep sequences. For example, if you create a new variable of lag 1 to lag 2 transitions (e.g., Kid1lag12trans), you can create a grid which displays three-step sequences in each cell by combining the state variable for the lag 0–1 transition on one axis with the lag 1–2 transition on the other axis (Kid1lag12trans with Kid2lag01trans). A step further, you can create cells of four-step sequences from a single-state variable (Kid1lag01trans with Kid1lag23trans). Although there are many instances in which this type of sequencing answers the research question of interest, these approaches may be overwhelming at first or even a bit cumbersome. One way to reduce the complexity a bit is to make use of the ordinality of your states. In the example we created in this chapter, this ordinality can be used in two ways, to examine relative change or cumulative changes in behavior.

With ordinal state variables, the change from one value to another is an equivalent quantitative change. Thus, we can say that a transition from 1 to 0 is equivalent to a change from 0 to -1. Thus, similar to the lag transitions above, we can examine the transitions in terms of relative change. In our example, we have two variables that reflect this approach, Kid1relchange and Kid2relchange. The new value at time t was computed as the original value at time t minus the original value at time $t-1$, beginning with the second event (the first event starts at zero). Essentially, this is a change score. Now, we have a sequence of incremental changes relative to the last

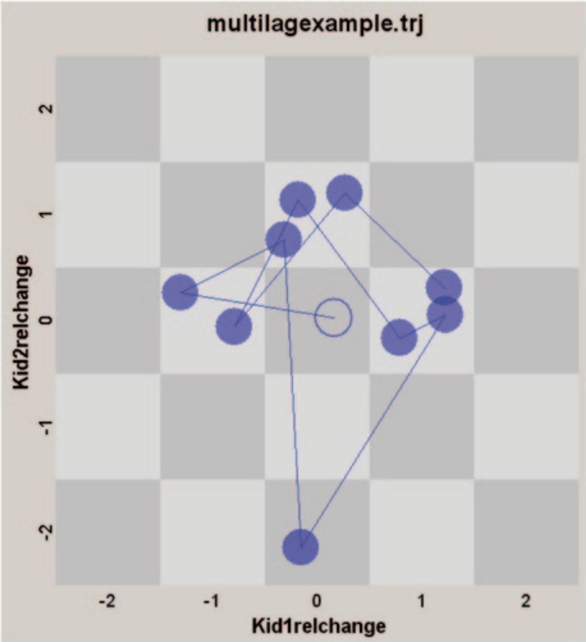

Fig. 7.7 State space grid of relative changes in Kid1 and Kid2 behavior

state (Fig. 7.7). In Fig. 7.7, trajectories in the upper right quadrant would indicate escalating positive (or decreasing negative) behavior, whereas trajectories in the lower left quadrant would indicate escalating negative (or decreasing positive) behavior. In this way, we can use a two-dimensional space to depict transitions that were visually evident on the state space grids from the raw data but that were not quantifiable without expanding the dimensionality.

Finally, using this transitional approach to ordinal sequences, we can make two-dimensional cumulative sum plots. Gottman et al. have used this approach to depict ordinal escalations and descalations of conflict behavior (Gottman 1979, 1993; Gottman and Levenson 1992). Recently, this was attempted with parent–child interactions as well (De Rubeis and Granic 2012). However, these approaches depict the time series in a traditional way, with time on the x-axis and a graph line moving left to right to depict each person's cumulative behavioral valence. State space grids can be used for this approach instead. In our current example, I have included two cumulative sum variables calculated as the value at time t plus the value at time $t-1$, Kid1cumchange and Kid2cumchange. This works because these are ordinal values and their addition retains the direction of the change. It also means that, unlike the relative change variables, sequences of repeating negative or positive changes compound. Figure 7.8 shows the cumulative sum plots in the traditional way as well as the corresponding state space grid. Visually, these are just two options for display. Computationally, however, displaying these cumulative sum plots in state space grids allows all sorts of analysis based on region selections and the exporting of measures from GridWare. For example, sometimes the final value is all that is

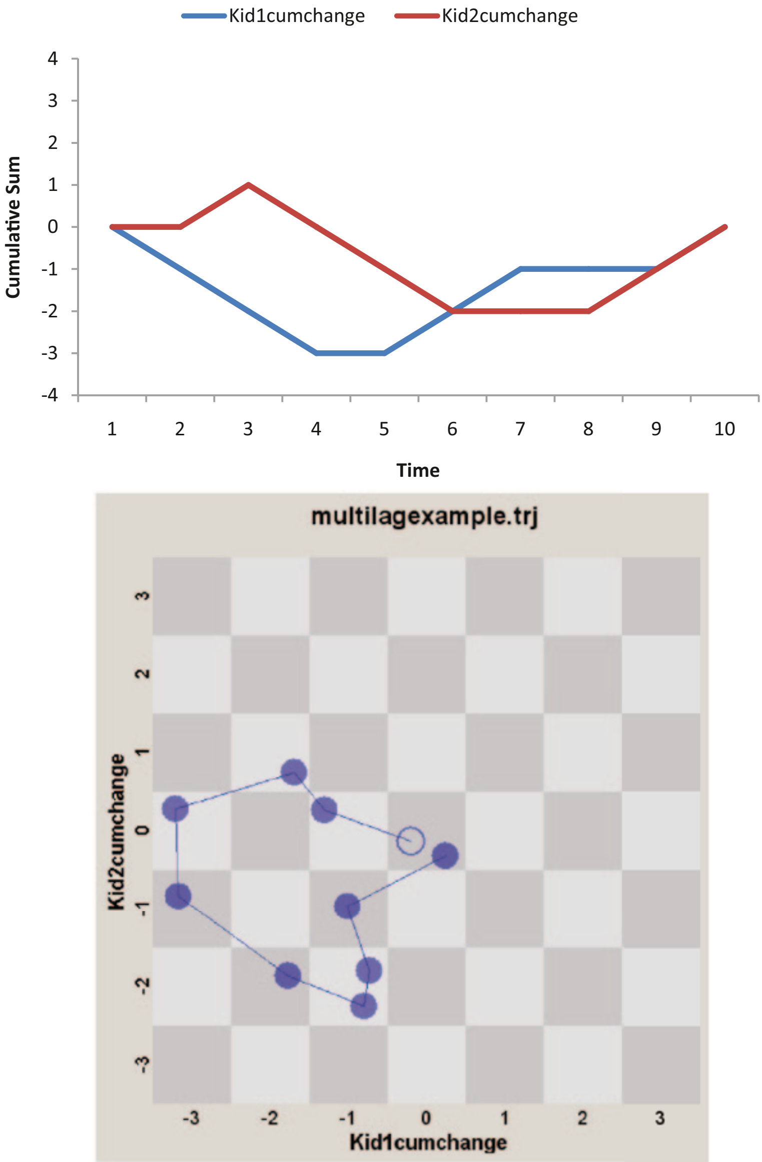

Fig. 7.8 Cumulative sum plots of example data. The top plot is the traditional time series display and the bottom plot is the state space grid method with the same data

analyzed in cumulative sum studies. If so, simply move the time slider to get to the final state before exporting. The added bonus with cumulative sum state space grids is that the degree of coordination between the two trajectories can be detected. The state space grid plot in Fig. 7.8 shows the classic conflict-resolution pattern that is less easily detected in the upper plot.

The take home message from this section should be that the state space can be manipulated in many ways to create multidimensional space. The analyses this enables are more complex but often fit the hypotheses and research questions of interest to developmental psychologists. With this new example you just created, you should be able to see the possibilities for your own work.

Using Continuous Variables

State space grids necessarily require categorical variables, variables with a relatively small number of discrete values. This creates a problem with continuous variables because the number of discrete possible values can be quite large. If you have continuous data, you may want to consider using other dynamic analyses as these often presume data of a continuous nature. Sometimes, however, the dynamics of these variables relative to other categorical variables is of interest. For example, in my newer research, I examine the concordance between psychophysiological indicators of autonomic nervous system activity (e.g., heart rate) and behavior and cognitions during emotionally stressful situations. Two of these measurement domains produce continuous measures and thus require alterations in order to analyze the dynamics via state space grids. Heart rate is an irregular time series that changes values each time a new heart beat is recorded. Thus, even though it is measured in milliseconds, the value changes somewhat discretely in time and then remains at that value until the next beat. Visually, on a traditional time series graph this would look similar to a series of stair steps going up and down as the heart rate changes over time, left to right. The other variable is the self report of the participant obtained through the use of a lever. The farther the participant pushes the lever forward, the more distressed they are. This variable is recorded in milliseconds and ranges from 0 to 100 with different values at the 0.001 scale. As these two variables are good examples of different kinds of continuous measures, I will use them to cover the options.

The self-report measure is more straightforward, so let us start with that. Here, the scale is arbitrary. We could have made the lever location correspond to −500 to 1,000 or 0 to a million. The choice for us here is to determine *what is a meaningful amount of change*. We could break up the 100 point scale into 100 categories, for example, one per unit. In this case, changes of .1 or less would not correspond to a change of event. That would give us a grid dimension of 100 categories. If whole-grid variability was the goal, then this arrangement would result in a high number of visits and transitions that may not be meaningful. At the other extreme, we could divide the 100 scale into three categories: low, medium, and high. This is the least complex and may better reflect the nature of the questions involved if all you are interested in are those times when self-reported distress is high. I suggest a compromise between those two extremes. Using ten categories (0–10, 10–20, etc.) still provides sufficient detail about the rates of changes and the similar but does not account for small-scale changes. However, with any new endeavor, it is best to explore first. Therefore, I recommend creating two or three solutions (e.g., categories of 1, 5, or 10 units) and compare the results you get from each project.

With many continuous measures, the issue of individual differences in baseline or reporting biases can arise. For example, with the self-reported distress lever, it is possible that some people recorded within a smaller range than others who used the entire spectrum from 0 to 100. Experimenter instructions, demonstrations, and practice are attempts to minimize that in this case, but it is still possible for participants to anchor their behavior differently, even though their experiences are comparable. For this reason, it is a possible benefit to rescale these continuous measures as within-subject standardized scores. Standardization of time series variables can be a useful technique in many situations. Essentially, this transformation rescales the series values into deviations from the mean of the series expressed in standard deviation units. The advantage of this approach is that now we can make categories that are interpretable but also on the same scale across subjects. Categories can be made from break points at 0.5 or 1 standard deviations, with the outer categories reflecting values of ±3 SD or greater. Thus, any continuous series can be broken into an eight-category state variable with values of −3 (or less), −3 to −2, −2 to −1, −1 to 0, 0–1, 1–2, 2–3, and 3 (or more). Using units of 0.5 standard deviations would result in a 14-category state variable. Again, the choice of how many categories depends on the nature of the variable, how frequently and dramatically the values change, and how it relates to the other state variables of interest in your project.

With a standardization approach, because 68 % of the values must occur between −1 and +1, every trajectory is centered in the state space. This is neither bad nor good in and of itself, but it does warrant your careful consideration with respect to your data and research questions. In some cases, you may choose to use between-subjects standardization for reasons that make sense for your data. The categorization process would be the same as the within-subject standardization, just using different mean and standard deviation values to standardize each value in the series.

With the physiological data, individual differences in resting levels means that there is typically a baseline value to which changes must be compared. If my resting heart rate is 25 beats per minute lower than yours, then a change of 10 beats per minute is not equivalent for each of us and, more importantly, there is the problem of comparisons within the same state space. Without some compensation, my absolute heart rate changes will be to the left of yours (if heart rate is the *x*-dimension), but this is only meaningful in the context of our different resting rates, not due to our different reactions to the task or event. There are several ways to adjust for baseline values with physiological data. One is a simple difference score, subtracting out baseline. Another is dividing that result by the baseline value to get a proportional change from baseline. Still, a third way is to standardize—within- or between-subjects as described above—with or without using the baseline levels. My preference is to use the proportional percent change from baseline. This is the best way to ensure that a change of 10 beats per minute, for example, is a proportionally greater change for someone with a baseline heart rate of 50 beats per minute than for someone with a resting heart rate of 80 beats per minute.

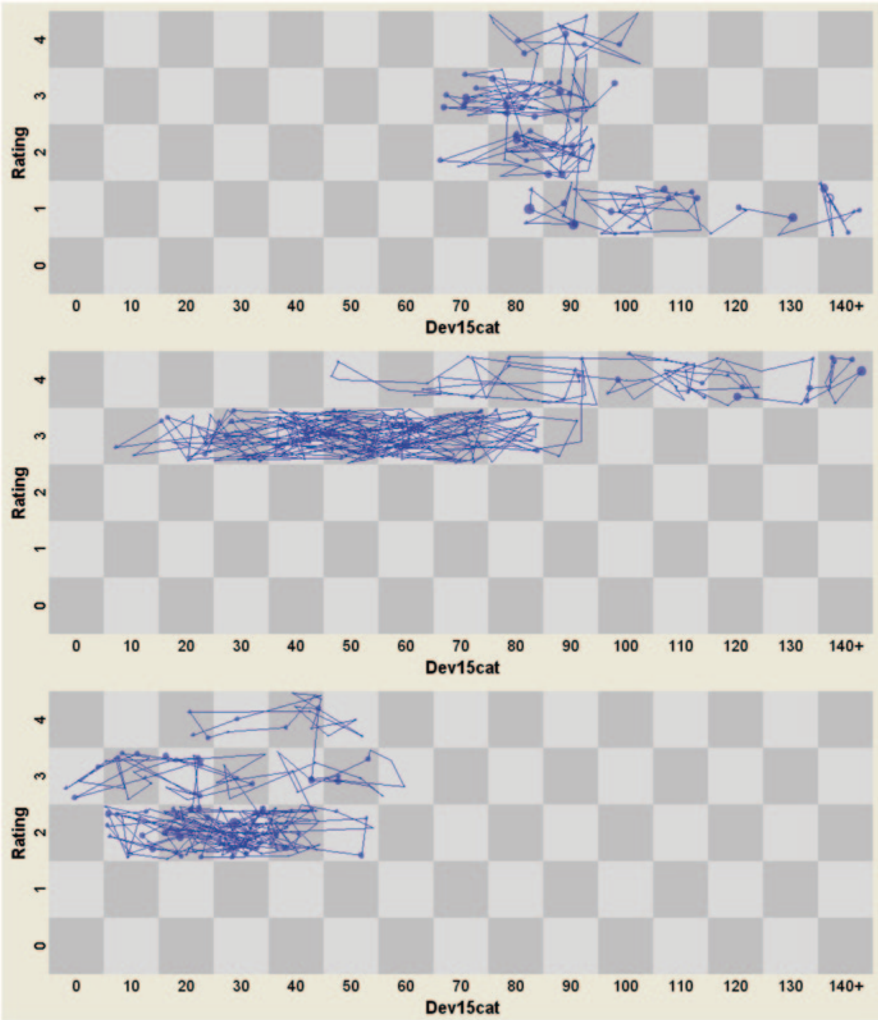

Fig. 7.9 Three examples of continuous measures broken up into categories for state space grid analysis. For each grid, the *x*-dimension is made up of categories of heart rate percent deviation from baseline; the *y*-dimension is the 0–100 self-report measure of distress in five 20-unit categories

Putting these options together, consider the examples in Fig. 7.9. These are the data from participants who gave an unrehearsed speech in the laboratory. They rated their distress as a continuous measure that recorded the relative position of a lever from 0 to 100. The values in those time series were categorized in five groups of 20 (0 = 0–20, 1 = 20–40, 2 = 40–60, 3 = 60–80, and 4 = 80–100). On the *x*-dimension are the percent deviations from baseline segmented into 15 categories of 10 % bins. The final bin category captured everything at 140 % deviation from baseline or greater (which did not happen often). The three examples here show three distinct patterns. In the top state space grid, this person rated herself as not very distressed at all while her heart rate climbed to the highest possible levels. This may indicate a process of suppression or even perhaps providing an inaccurate report. The second example shows a person who seems quite concordant across both domains. The third example shows someone whose physiology does not show that dramatic of a change but does report being highly distressed. This may indicate the overreaction of a catastrophizer or someone with a blunted physiological response.

These examples also expose a feature of continuous variables that has implications for visualization and measurement. Because these measures are often taken at high rates (e.g., 200 Hz), they have many values that can occur within the categories you have made. Thus, a final step with categorization processing is to eliminate redundant events (rows in the trajectory files). The squiggly nature of the trajectories plotted in Fig. 7.9 is due to this last step not yet being complete. That "squiggliness" of a high number of events per visit can occur in other instances as well (e.g., projects with many state variables). Thus, it is sometimes useful to separate your projects into simple two-state-variable projects, collapse the repeating rows, and conduct analyses specific to that state space.

Complementary Analyses

State space grid analysis is just one technique out of a growing field of methodologies based on dynamic systems or just the analysis of categorical time series. State space grids are an important tool but often it takes many tools to complete the understanding of the phenomenon at hand. In this section, I briefly review some techniques that are very good complements to state space grid analysis. This is certainly not an exhaustive list as there are many approaches out there (e.g., Agresti 2010; Valsiner et al. 2011), but the ones I describe below appear to hold the most immediate promise.

Continuous State Space Analysis

As described in this chapter, the analysis of continuous measures with state space grids is a compromise born out of the need to integrate categorical and continuous variables. Nonetheless, there is a wide range of tools that can analyze continuous dimensions of state space via mathematical formulae. Many of these are technically quite daunting for the average researcher in developmental psychology or many other areas of social science. They often require very long time series and a fidelity of measurement not easily achieved. However, there are some techniques that are accessible methods. One promising avenue is hierarchical state space analysis (Lodewyckx et al. 2011) that enables the search for attractors and other analyses. It is quite technical at this stage but the authors are preparing a user-friendly software application for general distribution. With this and other continuous state space tools, we can converge on a useful set of approaches to answer most any questions about system dynamics in psychology or other social sciences.

Sequential Analyses

Sequential analyses have been around for a long time and they are based on conditional probabilities. The core question is “given a particular state A, what is the probability that B will occur?” I recommend two books in particular by Roger Bakeman that provide an excellent introduction to the novice and clear explanation of the issues (Bakeman and Gottman 1997; Bakeman and Quera 1995). One of the reasons that state space grids initially held so much appeal for me was to overcome some of the limitations of these approaches. For one, the specification of a single-lagged event is required for the calculations, but often meaningful responses in interpersonal interactions are not immediate. Thus, the constraints of lag 1 relationships are relieved in state space grids. A second issue has to do with the enormous sensitivity of these conditional probabilities to the base rates of states and the length of the time series. This is one reason why transitional propensities, rather than conditional probabilities, are included in GridWare. Nevertheless, with lag grids, we return to some of the perennial constraints of sequential analysis and therefore it is recommended that you learn about these approaches if you are going to analyze lag grids.

Inertia

Recently, Kuppens et al. (2010, 2011) have examined emotional dynamics through inertia—the resistance to changing emotional states (Sheeber et al. 2011). With this approach, inertia has been operationalized at the individual level computed as an autoregressive correlation. Higher inertia (autocorrelation) indicates a degree of emotional rigidity in that it reflects getting stuck in emotional states. Indeed, the results from the inertia studies indicate greater problems (e.g., depression) for those with higher emotional inertia. The overlap and distinctiveness between dyadic rigidity and inertia is just starting to be explored and represents a fruitful avenue of exploration with state space grid analysis.

Identifying Multistep Sequences with T-Patterns

Magnusson and colleagues have been developing and advocating person-centered approaches for years. One analytical approach that is quite complementary to state space grids is the detection of hierarchical time patterns called T-patterns (Magnusson 2000). With this approach, a sequence of states is subjected to an algorithm to detect repeating sequences of length *n*, starting with the longest repeating sequence. It functions much like a hierarchical cluster analysis but incorporates time as well. In addition, it can test the occurrence of these repeating chains against chance to establish significance. In concert with state space grid analysis, T-patterns may reveal cyclic attractors of repeating sequences of events. The combination of these two approaches is a currently untapped potential.

Visualizing Simulation Data

To test the specific predictions of a theory, researchers often simulate data to model the processes and mechanisms hypothesized to underlie a phenomenon. With simulations, you can create hundreds or thousands of data series and estimate the reliability of theoretical assumptions. Sometimes these simulations result in data well suited for state space grids and can be visualized using GridWare. For example, in a study of peer interactions, Steenbeek and van Geert (2008) used agent-based modeling to derive play interaction patterns in dyads with children of rejected, average, and popular sociometric status. By selecting parameters of hypothesized variables, they created multiple simulations to demonstrate how observed patterns of peer behavior could arise. Some of the output variables included the emotions of the two children, and these could easily be input into state space grids for visualizing the interaction patterns. In general, any synchronized time series can be input for state space grid analyses and visualization, no matter how it is measured.

Recurrence Quantification Analysis

Many techniques are designed to detect the recurrence of events or sequence of events and thus have great potential for attractor analyses. In a variation of recurrence plots (e.g., Visual Recurrence Analysis), recurrence quantification analysis (RQA) is a robust method of quantifying the number and duration of recurrences in a dynamic system presented by its state space grid trajectory (Zbulit and Webber 1992). There are various software packages for conducting RQA (for comparisons, see Belaire-Franch and Contreras 2002). This approach is only starting to be applied in the social sciences (Stephen et al. 2009) and is an ideal complement to state space analysis.

Event History Analysis

I saved the best for last. Event history analysis (EHA) is a variation on survival analyses in which the events of interest can repeat. It uses both time and event information in a way that may be best suited for attractor analysis. The null model is that the trajectory will transition into a state as a function of time—the longer it has been since the last event, the more likely it becomes over time (note that this is the source of transitional propensities as described in Chap. 5). The improbability of this recurrence then can be interpreted as an attractor state. Moreover, EHA allows for time-varying covariates, so that not only you can model the repeating event of interest but also include other states that are changing at the same time as predictors of the repeating event. For example, Snyder et al. (2003) modeled children's anger displays using EHA using parent's anger as a time-varying covariate, reflecting the degree to which the mother's anger changed the transitional propensity of the child's anger. Thus, EHA is a powerful tool to assess the dynamic recurrence of system states within the dynamic context of the system itself. Understandably, there is a growing interest in this technique and a few options for how to conduct these analyses in Mplus (Muthen and Masyn 2005), Splus (Stoolmiller and Snyder 2006), and R (Mills 2011). The accessibility of these techniques will no doubt result in an increase in studies based on EHA and in concert with state space grid analysis.

Conclusions

The aim of this chapter was to give an idea of the range of possibilities for advanced state space grid analysis, from three-dimensional to EHA. The state space grid approach has only begun to be explored and these extensions represent the cutting-edge of this technique. To conclude, I have just a few more recommendations for how to approach these analyses. First, in most cases, I suggest doing whatever transformations or reformatting on one or two files first. Bring these files to the final arrangement and examine the result. This will save a lot of headaches because you will no doubt realize things in the final solution that will change the way you proceed from the beginning with the rest of the files. Second, get comfortable with some light "programming" such as Visual Basic macros. It is not that daunting anymore and just a little knowledge can go a long way. Of course, the alternative is to find a competent programmer to do it for you. But I still would advocate that, as the researcher, you understand the steps taken to transform your data. Third, when all your analyses are set up, go back to your source (e.g., observational video) and compare with what you see on the grids. It should make sense.

There are some remarkable discoveries ahead. In the next edition of this book, I expect this chapter to be twice as long. I look forward to including your results.

Epilogue

State space grids began as a solution to a very immediate problem. As Marc Lewis and Alex Lamey brainstormed about ways to reveal attractors in infant socioemotional behavior, they were simply trying to test a hypothesis on a small data set. Poetically, the state space grids idea self-organized and grew. It was a useful technique picked up by one person and then another. The reaction from venerable scholars and first year students was immediately positive and contagious. With the advent of GridWare, the technique became accessible to a large number of researchers. There were hundreds of software downloads within a short time. More and more studies were conducted using state space grids. Then came the requests for workshops and now there is a book to explain it all. From an immediate problem to a global solution, there is more growth ahead.

The range and quality of research presented here reflects the hard work of many people. Of course, the primary credit goes to Marc Lewis, Isabel Granic, and Alex Lamey who got this all started. They laid the groundwork that has made the quality and proliferation of state space grids possible. They made the ball, I just ran with it. Jerry Patterson and Mike Stoolmiller were early supporters who made supportive contributions to this work. Tom Dishion has been and continues to be one of the strongest supporters of state space grid analysis. Over the years, I have learned a great deal from those who invited me to do workshops—too many individuals to possibly list them all. Still, I would like to thank those at Arizona State, Wayne State, UCLA, York University and the universities of Michigan, Groningen, Pittsburgh, Arizona, Groningen, Utrecht, and Valle (Columbia). I have also been impressed with the number and quality of Master's and Doctoral theses using state space grids (e.g., Baetz 2003; Erickson 2010; Gardner 2004; Granic 2000; Guo 2011; Hao 2006; Herbers 2011; Hogue 2010; Hollenstein 2005; Howerter 2010; Kear 2011; Lavictoire 2010; Lee 2011; Litteljohn 2004; McCutcheon 2005; Murphy-Mills 2011; Rolling 2008; Sravish 2011; Williams 2008). This is the next generation of researchers who will take this technique further than I can currently imagine. I would also like to thank my current collaborators who are pushing with me to make state space grid analysis reach its fuller potential, especially Erika Lunkenheimer, Nick Allen, Peter Kuppens, Anna Lichtwarck-Aschoff, Meredith Chivers, and Emily Butler. Finally, various funding agencies made the state space research

T. Hollenstein, *State Space Grids,* DOI 10.1007/978-1-4614-5007-8,

I conducted possible, including the National Institute of Mental Health and the Tri-council of Canada: Social Sciences and Humanities Research Council, Canadian Institutes of Health, and the Natural Sciences and Engineering Research Council.

It is February 2012 as I complete this book while on sabbatical in Melbourne, Australia. I began writing in a cabin among the fiery autumn foliage of Lac Bevan in the mountains of Quebec and completed most while under the dome in the State Library of Victoria; auspicious locations, to say the least. The completion of the book is much like an end to the first chapter of the state space grid story, covering a dozen years of contributions. The next installment covering the next dozen years will certainly be filled with new directions and new domains. If you are reading this, then you will probably be one of the people making an impact on the next chapter. I look forward to learning what you have discovered.

References

Agresti, A. (2010). *Analysis of ordinal categorical data*. Hoboken: Wiley.

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In B. N. Petrov & P. Czaki (Eds.), *2nd international symposium on information theory* (pp. 267–281). Budapest: Akademiai Kiado.

Ashby, W. R. (1947). Principles of the self-organizing dynamic system. *Journal of General Psychology, 37,* 125–128.

Baetz, C. P. (2003). *The relation between false belief understanding and the development of shame: A dynamic systems analysis* (Doctoral Thesis). ProQuest Dissertations and Theses. http://search.proquest.com/docview/305240775?accountid=6180. Accessed 12 Feb 2012.

Bakeman, R., & Gottman, J. M. (1997). *Observing interaction: An introduction to sequential analysis*. New York: Cambirdge.

Bakeman, R., & Quera, V. (1995). *Analyzing interaction: sequential analysis with SDIS & GSEQ*. New York: Cambridge.

Belaire-Franch, J., & Contreras, D. (2002). Recurrence plots in nonlinear time series analysis: Free software. *Journal of Statistical Software, 7,* 1–18.

Bergman, L. R., & Magnusson, D. (1997). A person oriented approach in research on developmental psychopathology. *Development and Psychopathology, 9,* 291–319.

Bergman, L. R., Magnusson, D., & El-Khouri, B. M. (2003). *Studying individual development in an interindividual context: A person-oriented approach*. Mahwah: Lawrence Erlbaum Associates.

Bronfenbrenner, U. (1979). *The ecology of human development: Experiments by nature and design*. Cambridge: Harvard University Press.

Celeux, G., & Soromenho, G. (1996). An entropy criterion for assessing the number of clusters in a mixture model. *Journal of Classification, 13,* 195–212.

Cohen, J. & Stewart, I. (1994). *The Collapse of Chaos: discovering simplicity in a complex world.* Penguin Books

De Rubeis, S., & Granic, I. (2012). Understanding treatment effectiveness for aggressive youth: The importance of regulation in mother–child interactions. *Journal of Family Psychology, 26,* 66–75.

de Waal, F. B. M. (1999). The end of nature versus nurture. *Scientific American, 281,* 94–99.

Diacu, F. (1992). *Singularities of the N-body problem: An introduction to celestial mechanics*. Montreal: Publications CRM.

Dishion, T. J., Nelson, S. E., Winter, C., & Bullock, B. (2004). Adolescent friendship as a dynamic system: Entropy and deviance in the etiology and course of male antisocial behavior. *Journal of Abnormal Child Psychology, 32,* 651–663.

Erickson, K. T. (2010). *State space grids: First application of a novel methodology to examine coach-athlete interactions in competitive youth sport* (Master's Thesis). ProQuest Dissertations and Theses. http://search.proquest.com/docview/760902118?accountid=6180. Accessed 12 Feb 2012.

T. Hollenstein, *State Space Grids,* DOI 10.1007/978-1-4614-5007-8,

Erickson, K., Cote, J., Hollenstein, T., & Deakin, J. (2011). Examining coach–athlete interactions using state space grids: An observational analysis in competitive youth sport. *Psychology of Sport and Exercise, 12,* 645–654.

Ford, D. H., & Lerner, R. M. (1992). *Developmental systems theory: An integrative approach*. Newbury Park: Sage.

Fogel, A. (1993). *Developing through relationships: Origins of communication, self and culture*. Chicago: University of Chicago Press.

Fogel, A., & Thelen, E. (1987). Development of early expressive and communicative action: Reinterpreting the evidence from a dynamic systems perspective. *Developmental Psychology, 23,* 747–761.

Gardner, B. C. (2004). *A dynamic systems exploration of affective self-organization processes during marital interaction: Attractors, perturbations, and phase transitions* (Doctoral Thesis). ProQuest Dissertations and Theses. http://search.proquest.com/docview/305132781?accountid=6180. Accessed 12 Feb 2012.

Gottlieb, G. (1991). Epigenetic systems view of human development. D*evelopmental Psychology, 27,* 33–34.

Gottlieb, G. (2007). Probabilistic epigenesis. *Developmental Science, 10,* 1–11.

Gottlieb, G., & Willoughby, M. T. (2006). Probabilistic epigenesis of psychopathology. In D. Cicchetti & D. J. Cohen (Eds.), *Developmental psychopathology: Theory and method* (vol. 1, 2nd ed.). New York: Wiley.

Gottman, J. M. (1979). *Marital interaction: Experimental investigations*. New York: Academic Press.

Gottman, J. M. (1993). The roles of conflict engagement, escalation, and avoidance in marital interaction: A longitudinal view of five types of couples. *Journal of Consulting and Clinical Psychology, 61,* 6–15.

Gottman, J. M., & Levenson, R. W. (1992). Marital processes predictive of later dissolution: Behavior, physiology and health. *Journal of Personality and Social Psychology, 63,* 221–233.

Gottman, J. M., McCoy, K., Coan, J., & Collier, H. (1996). *The specific affect coding system (SPAFF) for observing emotional communication in marital and family interaction*. Mahwah: Erlbaum.

Gottman, J. M., Murray, J. D., Swanson, C. C., Tyson, R., & Swanson, K. R. (2002). *The mathematics of marriage: Dynamic nonlinear models*. London: MIT Press.

Granic, I. (2005). Timing is everything: Developmental psychopathology from a dynamic systems perspective. *Developmental Review, 25*(3–4), 386–407.

Granic, I., & Hollenstein, T. (2003). Dynamic systems methods for models of developmental psychopathology. *Development and Psychopathology, 15*(3), 641–669.

Granic, I., & Hollenstein, T. (2006). A survey of dynamic systems methods for developmental psychopathology. In D. Cicchetti & D. J. Cohen (Eds.), *Developmental psychopathology* (pp. 889–930). New York: Plenum Press.

Granic, I., & Lamey, A. V. (2002). Combining dynamic systems and multivariate analyses to compare the mother–child interactions of externalizing subtypes. *Journal of Abnormal Child Psychology, 30*(3), 265–283.

Granic, I., & Patterson, G. R. (2006). Toward a comprehensive model of antisocial development: A dynamic systems approach. *Psychological Review, 113*(1), 101–131.

Granic, I., Hollenstein, T., Dishion, T. J., & Patterson, G. R. (2003). Longitudinal analysis of flexibility and reorganization in early adolescence: A dynamic systems study of family interactions. *Developmental Psychology, 39*(3), 606–617.

Granic, I., O'Hara, A., Pepler, D., & Lewis, M. D. (2007). A dynamic systems analysis of parent–child changes associated with successful "real-world" interventions for aggressive children. *Journal of Abnormal Child Psychology, 35*(5), 845–857.

Granic, I., Meusel, L., Lamm, C., Woltering, S., & Lewis, M. (2012). Flexibility in parent–child interactions is associated with unique cortical mechanisms of emotion regulation. *Development and Psychopathology, 24*, 1019–1029.

Guastello, S. J. (1995). *Choas, catastrophe, and human affairs: Applications of nonlinear dynamics to work, organizations, and social evolution*. Mahwah: Lawrence Erlbaum Associates.

Guo, Y. (2011). *A dynamic systems approach to co-emotion regulation in preschoolers and mothers in the context of strange situation: Predictors and developmental consequences* (Doctoral Thesis). ProQuest Dissertations and Theses. http://search.proquest.com/docview/905268744?accountid=6180. Accessed 12 Feb 2012.

Hao, H. (2006). *Early expertise development and transitions in a nonuniversal domain: A multimethod approach* (Doctoral Thesis). ProQuest Dissertations and Theses, p. 116. http://search.proquest.com/docview/304985652?accountid=6180. Accessed 12 Feb 2012.

Haken, H. (1977). *Synergetics: An introduction*. Berlin: Springer.

Hartelman, P., van der Maas, H., & Molenaar, P. (1998). Detecting and modeling developmental transitions. *British Journal of Developmental Psychology. Special Issue: Developmental Transitions, 16,* 97–122.

Heath, R. A. (2000). *Nonlinear dynamics: Techniques and applications in psychology*. Mahwah: Erlbaum.

Herbers, J. E. (2011). *Parent–child relationships in young homeless families: Co-regulation as a predictor of child self-regulation and school adjustment* (Doctoral Thesis). ProQuest Dissertations and Theses. http://search.proquest.com/docview/898774094?accountid=6180. Accessed 12 Feb 2012.

Hogue, U. T. (2010). *Childhood externalizing behaviour and parent–child interaction: Associations among cortisol reactivity, emotion, child and parent psychopathology* (Doctoral Thesis). ProQuest Dissertations and Theses. http://search.proquest.com/docview/816795584?accountid=6180. Accessed 12 Feb 2012.

Hollenstein, T. (2007). State space grids: Analyzing dynamics across development. *International Journal of Behavioral Development, 31,* 384–396.

Hollenstein, T. (2011a). Twenty years of dynamic systems approaches to development: Significant contributions, challenges, and future directions. *Child Development Perspectives, 5,* 256–259.

Hollenstein, T. (2011b). Using state space grids for understanding processes of change and stability in adolescence. In E. S. Kunnen (Ed.), *A dynamic systems approach to adolescent development*. London: Psychology Press.

Hollenstein, T. (2012). The dynamic systems perspective: What is the system? In L. C. Mayes & M. Lewis (Eds.), *The Cambridge handbook of environment in human development*. New York: Cambridge University Press.

Hollenstein, T., & Lewis, M. D. (2006). A state space analysis of emotion and flexibility in parent–child interactions. *Emotion, 6,* 663–669.

Hollenstein, T., Lichtwarck-Aschoff, A., & Potworowski, G. (2012). A model of socioemotional flexibility at three time scales. *Emotion Review*.

Hollenstein, T., Granic, I., Stoolmiller, M., & Snyder, J. (2004). Rigidity in parent–child interactions and the development of externalizing and internalizing behavior in early childhood. *Journal of Abnormal Child Psychology, 32,* 595–607.

Howerter, A. (2010). *A dynamic system perspective on interpersonal emotion regulation* (Doctoral Thesis). ProQuest Dissertations and Theses. http://search.proquest.com/docview/746617344?accountid=6180. Accessed 12 Feb 2012.

Hsu, H., & Fogel, A. (2001). Infant vocal development in a dynamic mother–infant communication system. *Infancy, 2,* 87–109.

Hubler, D. (2009). *Exploring finances, emotions, and couple relations using the family stress model and stress spilover model* (Master's Thesis). ProQuest Dissertations and Theses. http://search.proquest.com/docview/305086979?accountid=6180. Accessed 12 Feb 2012.

Isen, A. M. (2000). Some perspectives on positive affect and self-regulation. *Psychological Inquiry, 11,* 184–187.

Johnson, S. L., & Nowak, A. (2002). Dynamical patterns in bipolar depression. *Personality and Social Psychology Review, 6,* 380–387.

Kauffman, S. (1995). *At Home in the Universe: The Search for Laws of Self-Organization and Complexity*. Oxford University Press

Kear, E. (2011). *Parent–adolescent relationships: anticipations and dyadic interactions during the transition to high school* (Master's Thesis).

Kelso, J. A. S. (1995). *Dynamic patterns: The self-organization of brain and behavior*. Cambridge: MIT Press.

Koval, P., & Kuppens, P. (2011). Changing emotion dynamics: Individual differences in the effects of anticipatory social stress on emotional inertia. *Emotion, 12*, 256–267.

Kuppens, P., Allen, N. B., & Sheeber, L. (2010). Emotional inertia and psychological maladjustment. *Psychological Science, 21,* 984–991.

Kuppens, P., Sheeber, L., Yap, M., Whittle, S., Simmons, J. G., & Allen, N. B. (2011). Emotional inertia prospectively predicts the onset of depressive disorder in adolescence. *Emotion, 12*, 283–289.

Lamey, A., Hollenstein, T., Lewis, M. D., & Granic, I. (2004). *GridWare (Version 1.1)* [Computer software]. http://statespacegrids.org.

Lanteigne, D., Glozman, J., & Hollenstein, T. (2011, April). *The development of an observational coding system for self-conscious affect in adolescence*. Poster presented at the biennial meeting of the Society for Research in Child Development, Montreal, QC.

Lanteigne, D., Flynn, J. J., Eastabrook, J., & Hollenstein, T. (2012). Discordant patterns among emotional experience, arousal, and expression in adolescence: Relations with emotion regulation and internalizing problems. *Canadian Journal of Behavioural Science*.

Larsen, R., & Csikszentmihalyi, M. (1983). The experience sampling method. *New Directions for Methodology of Social & Behavioral Science, 15,* 41–56.

LaVictoire, L. A. (2010). *Affective dynamics of rejected children in triadic peer interactions in early childhood* (Master's Thesis). ProQuest Dissertations and Theses. http://search.proquest.com/docview/853332224?accountid=6180. Accessed 12 Feb 2012.

Lavictoire, L., Snyder, J., Stoolmiller, M., & Hollenstein, T. (2012). Affective dynamics in triadic peer interactions in early childhood. *Nonlinear Dynamics in Psychology and the Life Sciences, 16*, 293–312.

Lee, V. (2011). *Flexibility in parent–child interactions: The application of dynamic systems methodology to dyadic processes in children with behaviour problems* (Doctoral Dissertation). Electronic Thesis and Dissertation Repository. Paper 352. http://ir.lib.uwo.ca/etd/352

Lewis, M. D. (2000a). Emotional self-organization at three time scales. In M. D. Lewis & I. Granic (Eds.), *Emotion, development, and self-organization: Dynamic systems approaches to emotional development* (pp. 37–69). New York: Cambridge University Press.

Lewis, M. D. (2000b). The promise of dynamic systems approaches for an integrated account of human development. *Child Development, 71,* 36–43.

Lewis, M. D. (2005). Bridging emotion theory and neurobiology through dynamic systems modeling (target article). *Behavioral and Brain Sciences, 28,* 105–130.

Lewis, M. D. (2011). Dynamic systems approaches: Cool enough? Hot enough? *Child Development Perspectives, 5,* 279–285.

Lewis, M. D., & Liu, Z. (2011). Three time scales of neural self-organization underlying basic and nonbasic emotions. *Emotion Review, 3,* 1–8.

Lewis, M. D., Lamey, A. V., & Douglas, L. (1999). A new dynamic systems method for the analysis of early socioemotional development. *Developmental Science, 2,* 457–475.

Lewis, M. D., Zimmerman, S., Hollenstein, T., & Lamey, A. V. (2004). Reorganization in coping behavior at 1 1/2 years: Dynamic systems and normative change. *Developmental Science, 7,* 56–73.

Lewis, M. D., Granic, I., & Lamm, C. (2006). Neurocognitive mechanisms of emotion regulation and behavioral flexibility in aggressive children. *Annals of the New York Academy of Sciences 1094*, 164–177.

Lichtwarck-Aschoff, A., van Geert, P., Bosma, H., & Kunnen, S. (2008). Time and identity: A framework for research and theory formation. *Developmental Review, 28,* 370–400.

Lichtwarck-Aschoff, A., Kunnen, S., & van Geert, P. (2009). Here we go again. A dynamic systems perspective on emotional rigidity across parent–adolescent conflicts. *Developmental Psychology, 45,* 1364–1375.

Litteljohn, A. (2004). *Developmental and individual correlates of mother–daughter attunement in early adolescence* (Master's Thesis). ProQuest Dissertations and Theses. http://search.proquest.com/docview/305067171?accountid=6180. Accessed 12 Feb 2012.

Lodewyckx, T., Tuerlinckx, F., Kuppens, P., Allen, N. B., & Sheeber, L. (2011). A hierarchical state space approach to affective dynamics. *Journal of Mathematical Psychology, 55,* 68–83.

Lunkenheimer, E. S., Olson, S. L., Hollenstein, T., Sameroff, A., & Winter, C. (2011). Dyadic flexibility and positive affect in parent–child coregulation and the development of childs behavior problems. *Development and Psychopathology, 23,* 577–591.

Mills, M. (2011). *Introducing survival and event history analysis*. London: Sage.

Magnusson, D. (1999). Holistic interactionism: A perspective for research on personality development. In L. A. Pervin and O. John (Eds.), *Handbook of personality: Theory and research* (pp. 219–247). New York: Guilford Press.

Magnusson, M. S. (2000). Discovering hidden time patterns in behavior: T-patterns and their detection. *Behavior Research Methods, Instruments and Computers, 32,* 93–110.

Magnusson, D. (2003). The person approach: Concepts, measurement models, and research strategy. *New Directions for Child and Adolescent Development, 101,* 3–23.

Martin, C. L., Fabes, R. A., Hanish, L. D., & Hollenstein, T. (2005). Social dynamics in the preschool. *Developmental Review, 25,* 299–327.

Matthews, A., & McLeod, C. (1985). Selective processing of threat cues in anxiety states. *Behavior Research and Therapy, 23,* 563–569.

McCarthy, A., Hollenstein, T., Muir, D., & Lee, K. (2007, April). *Children's eye gaze pattern during thinking and its relation to their knowledge about thinking-related gaze behavior*. Poster presented at the biennial meeting of the Society for Research in Child Development, Boston, MA.

McCutcheon, G. (2005). *Rigidity in mother–infant dyads: A state space analysis of emotion regulation* (Doctoral Thesis). ProQuest Dissertations and Theses. http://search.proquest.com/docview/276504063?accountid=6180. Accessed 12 Feb 2012.

Moskowitz, D. S., & Hershberger, S. L. (2002). *Modeling intraindividual variability with repeated measures data: Methods and applications*. Mahwah: Lawrence Erlbaum Associates.

Murphy-Mills, J. (2011). *An examination of coach–athlete interactions in a model sport program for athletes with disabilities* (Master's Thesis). http://qspace.library.queensu.ca/handle/1974/6588. Accessed 12 Feb 2012.

Muthén, B. (2001). Latent variable mixture modeling. In G. A. Marcoulides & R. E. Schumacker (Eds.), *New developments and techniques in structural equation modeling* (pp. 1–33). Mahwah: Erlbaum.

Muthen, B., & Masyn, K. (2005). Discrete-time survival mixture analysis. *Journal of Educational and Behavioral Statistics, 30,* 27–58.

Nylund, K. L., Asparouhov, T., & Muthén, B. O. (2007). Deciding on the number of classes in latent class analysis and growth mixture modeling: A Monte Carlo simulation study. *Structural Equation Modeling, 14,* 535–569.

Oyama, S., Griffiths, P. E., & Gray, R. D. (2001). *Cycles of contingency: Developmental systems and evolution*. Boston: MIT Press.

Overton, W. F. (2007). A coherent metatheory for dynamic systems: Relational organicism-contextualism. *Human Development, 50,* 154–159.

Piaget, J. (1952). *The origins of intelligence in children*. New York: International Universities Press.

Pinker, S. (2004). Why nature and nurture won't go away. *Daedalus, 133,* 5–17.

Prigogine, I. (1997). *The end of certainty*. New York: The Free Press.

Richters, J. E. (1997). The Hubble hypothesis and the developmentalist's dilemma. *Development and Psychopathology, 9,* 193–229.

Rolling, E. S. (2008). *Attachment styles and negative affect: Identifying underlying processes* (Master's Thesis). ProQuest Dissertations and Theses. http://search.proquest.com/docview/304385437?accountid=6180. Accessed 12 Feb 2012.

Ruhland, R., & van Geert, P. (1998). Jumping into syntax: Transitions in the development of closed class words. *British Journal of Developmental Psychology, 16,* 65–95.

Sameroff, A. J. (1983). Developmental systems: Contexts and evolution. In W. Kessen (Ed.), *History, theory, and methods* (pp. 237–294, Vol. 1, 4th ed.). New York: Wiley.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics, 6,* 461–464.

Sclove, L. S. (1987). Application of model-selection criteria to some problems in multivariate analysis. *Psychometricka, 52,* 333–343.

Shannon, C. E., & Weaver, W. (1949). *The mathematical theory of communication*. Urbana: University of Illinois Press.

Sheeber, L., Kuppens, P., Shortt, J. W., Katz, L. F., Davis, B., & Allen, N. B. (2011). Depression is associated with the escalation of adolescents' dysphoric behavior during interactions with parents. *Emotion.* Advance online publication. doi: 10.1037/a0025784

Silk, J. S., Forbes, E. E., Whalen, D. J., Jakubcak, J. L., Thompson, W. K., Ryan, N. D., Axelson, D. A., Birmaher, B., & Dahl, R. E. (2011). Daily emotional dynamics in depressed youth: A cell phone ecological momentary assessment study. *Journal of Experimental Child Psychology, 110,* 241–257.

Snyder, J., Stoolmiller, M., Wilson, M., & Yamamoto, M. (2003). Child anger regulation, parental responses to children's anger displays, and early child antisocial behavior. *Social Development, 12*(3), 335–360.

Spencer, J. P., Clearfield, M., Corbetta, D., Ulrich, B., Buchanan, P., & Schoner, G. (2006). Moving toward a grand theory of development: In memory of Esther Thelen. *Child Development, 77*(6), 1521–1538.

Sravish, A. V. (2011). Dyadic flexibility during the face-to-face still-face paradigm: *Analysis of infant-caregiver interactions within a dynamic systems framework using the state space grid analytic technique* (Master's Thesis). http://search.proquest.com/docview/873577573?accountid=6180. Accessed 12 Feb 2012.

Steenbeek, H., & van Geert, P. (2005). A dynamic systems model of dyadic interaction during play of two children. *European Journal of Developmental Psychology, 2,* 105–145.

Steenbeek, H. W., & van Geert, P. L. C. (2008). The empirical validation of a dynamic systems model of interaction: Do children of different sociometric statuses differ in their dyadic play interactions? *Developmental Science, 11,* 253–281.

Stoolmiller, M., & Snyder, J. (2006). Modeling heterogeneity in social interaction processes using multilevel survival analysis. *Psychological Methods, 11*(2), 164–177.

Thayer, J. F., Ahs, F., Fredrikson, M., Sollers, J. J., & Wagner, T. D. (2012). A meta-analysis of heart rate variability and neuro-imaging studies: Implications for heart rate variability as a marker of stress and health. *Neurosceince and Biobehavioral Reviews, 36,* 747–756.

Thelen, E., & Smith, L. B. (1994). *A dynamic systems approach to the development of cognition and action*. Cambridge: Bradford/MIT Press.

Thelen, E., & Smith, L. B. (2006). Dynamic systems theories. In W. Damon (Series Ed.) & R. M. Lerner (Vol. Ed.), *Handbook of child psychology: Vol. 1. Theoretical models of human development* (6th ed., pp. 258–312). New York: Wiley.

Thelen, E., & Ulrich, B. D. (1991). Hidden skills: A dynamic systems analysis of treadmill stepping during the first year. *Monographs of the Society for Research in Child Development, 56*(1, Serial No. 223).

Thelen, E., Schöner, G., Scheier, C., & Smith, L. B. (2001). The dynamics of embodiment: A field theory of infant perseverative reaching. *Behavioral and Brain Sciences, 24*(1), 1–86.

Thompson, E. (2007). *Mind in Life: biology, phenomenology, and the sciences of mind*. Cambridge: Belknap Press.

Tukey, J. W. (1977). Exploratory data analysis. Reading: Addison-Wesley.

Valsiner, J., Molenaar, P., Lyra, M., & Chaudhary, N. (2011). *Dynamic process methodology in the social and developmental sciences*. New York: Springer.

van der Maas, H. L. J. (1995). Beyond the metaphor? *Cognitive Development, 10,* 621–642.

van der Maas, H., & Molenaar, P. (1992). Stagewise cognitive development: An application of catastrophe theory. *Psychological Review, 99,* 395–417.

van Geert, P. (1994). *Dynamic systems of development: Change between complexity and chaos*. New York: Prentice Hall/Harvester Wheatsheaf.

van Geert, P. (1998). A dynamic systems model of basic developmental mechanisms: Piaget, Vygotsky, and beyond. *Psychological Review, 105,* 634–677.

van Geert, P. & Steenbeek, H. (2005). Explaining after by before: Basic aspects of a dynamic systems approach to the study of development. *Developmental Review, 25,* 408–442.

van Geert, P. & van Dijk, M. (2002). Focus on variability: New tools to study intraindividual variability in developmental data. *Infant Behavior & Development, 25,* 340–374.

von Bertalanffy, L. (1968). *General system theory: Foundations, development, applications*. New York: George Braziller.

Weiner, N. (1950). *The human use of human beings: Cybernetics and society*. Boston: Houghton Mifflin.

Williams, D. (2007). *A dynamic systems analysis of adult attachment styles, affect, flexibility, and real time marital interaction* (Master's Thesis). ProQuest Dissertations and Theses. http://search.proquest.com/docview/304744610?accountid=6180. Accessed 12 Feb 2012.

Wimmers, R. H., Savelsbergh, G. J. P., Beek, P. J., & Hopkins, B. (1998). Evidence for a phase transition in the early development of prehension. *Developmental Psychobiology, 32,* 235–248.

Witherington, D. C. (2007). The dynamic systems approach as metatheory for developmental psychology. *Human Development, 50*(2–3), 127–153.

Zbulit, J. P., & Webber, C. L. (1992). Embeddings and delays as derived from quantification of recurrence plots. *Physics Letters A, 171,* 199–203.

Index

T. Hollenstein, *State Space Grids,* DOI 10.1007/978-1-4614-5007-8,