Modeling and Simulation of Large Biological, Information and Socio-Technical Systems: An Interaction Based Approach

Chris Barrett Stephen Eubank Madhav Marathe

Virginia Polytechnic Institute and State University, Blacksburg Virginia.

Summary We describe an interaction based approach for computer modeling and simulation of large integrated biological, information, social and technical (BIST) systems[1] Examples of such systems are urban regional transportation systems, the national electrical power markets and grids, gene regulatory networks, the world-wide Internet, infectious diseases, vaccine design and deployment, theater war, etc. These systems are composed of large numbers of interacting human, physical, informational and technological components. These components adapt and learn, exhibit perception, interpretation, reasoning, deception, cooperation and non-cooperation, and have economic motives as well as the usual physical properties of interaction.

The theoretical foundation of our approach consists of two parts: (i) mathematics of complex interdependent dynamic networks, and (ii) mathematical and computational theory of a class of finite discrete dynamical systems called **Sequential Dynamical Systems** (SDSs). We then consider engineering principles based on such a theory. As with the theoretical foundation, they consist of two basic parts: (i) Efficient data manipulation, including synthesis, integration, storage and regeneration and (ii) high performance computing oriented system design, development and implementation. The engineering methods allow us to specify, design, and analyze simulations of extremely large systems and implement them on massively parallel architectures. As an illustration of our approach, an interaction based computer modeling and simulation framework to study very large interdependent societal infrastructures is described.

1 Introduction

This chapter considers an interaction based approach for modeling and simulation of large scale integrated biological, information, social and technical (henceforth referred to as BIST) systems. BIST systems consist of a large number of interacting

[1] To appear as a book chapter in the book titled *Interactive Computation: the New Paradigm*, Goldin, Smolka, Wegner, Editors, ESptringer Verlag, Sept. 2006.

physical, biological, technological, informational and human/societal components whose **global system** properties are a result of interactions among representations of **local** system **elements**. Examples of such systems are urban regional transportation systems, national electrical power markets and grids, the Internet, peer to peer networks, adhoc communication and computing systems, gene regulatory networks, public health, etc. The complicated interdependencies and interactions are inherent within and among constituent BIST systems. This is exemplified by the recent cascading failure of the electric grid in the northeastern United States. Failure of the grid led to cascading effects that slowed down Internet traffic, closed down financial institutions and disrupted the transportation and telecommunication systems.

In the past, mathematical models based on differential equations have often been used to model complex physical and social systems. Although such models are valuable in terms of providing simple first order explanations, they are not particularly useful in providing a generative computer model or a causal explanation of the associated dynamic phenomena. For instance, epidemiologists have traditionally used coupled differential rate equation based models on completely mixed populations to understand the spread of diseases. These simple models provide a good prediction for a number of important epidemiological parameters such as number of sick, infected and recovered individuals in a population. Nevertheless, such epidemiological models have a number of well known shortcomings. They include: an adhoc value of the reproduction number, the inability to predict anything about the early phase of disease spread, and an inability to account for spatial and demographic diversity in urban populations. Even more important, the models do not provide any causal explanation nor do they lead to a generative computational model. As a result, questions such as identifying potential individuals that can be vaccinated to contain the epidemic are very hard to analyze; see [22, 33, 46] for additional discussion.

Here, we describe an interaction based approach for modeling and simulation of BIST systems. The approach uses an endogenous representation of individual agents together with explicit interaction between these agents to generate and represent the causal ecologies in such systems. The approach was developed over the last 12 years by our group and provides a common framework for three seemingly diverse areas: (i) representation and analysis of large scale distributed BIST systems, (ii) next generation computing architectures and (iii) associated distributed information and data integration architectures.

The interaction-based approach is based on a mathematical and computational discrete dynamical systems theory called Sequential Dynamical System (SDS). SDSs provide a formal basis for describing complex simulations by composing simpler ones. They are a new class of discrete, finite dynamical systems and emphasize questions of what is being computed by systems of interacting elements, as opposed to the traditional approach of how *hard* it is to compute a given procedure or class. Nevertheless, a traditional Turing machine based approach is used for characterizing the computational complexity of the interacting elements.

We complement the theoretical discussion by describing **Simfrastructure**: a practical microscopic interaction-based modeling framework to study very large interdependent societal infrastructures formed by the interaction between the built urban infrastructure and spatial movement patterns of individuals carrying out their day-to-day activities. **Simfrastructure** has been used to model extremely large infrastructures consisting of millions of interacting agents consisting of more than 10 million individual elements. For example, the transportation module within **Simfrastructure** can represent every individual in the Chicago region at a temporal resolution of 1 second, and a spatial resolution of approximately 7 meters. This region spans approximately 250 square miles and has more than 400 counties. There are more than 9 million individuals taking roughly 25 million trips each day. The time varying social contact network consists of more than 25 million edges and vertices. The size, scope and multiple time scales of system representation naturally motivates a high performance computing implementation and requires new engineering design principles. Individual modules of this system routinely run on clusters comprised of 128 nodes; several of the individual simulations are also being executed on 1000+ node systems.

1.1 Relationship to Interactive Computing

There are at least two reasons why the topic of computer modeling and simulation of large BIST systems is pertinent to interactive computation. First, as discussed above, interaction based computer models are natural and the only way to represent and comprehend the complex dynamics of many BIST systems. In the past, computer simulation of physical phenomenon has been a key driver in the development of current high performance computing systems. Our view is that interaction based modeling and simulation of BIST systems will serve as a key driver for the development of next generation interactive computing platforms. Second, and perhaps more pertinent to this book, we believe that an interaction based modeling of BIST systems will yield new mathematical and computational techniques that advance the state of the art of interactive computation. Recently, computer scientists have proposed automata theoretic models, programming languages, and calculi that attempt to treat interaction, as an atomic element of computation. Several chapters in the book address these topics in detail. BIST systems naturally display many attributes of interactive computing such as providing a service rather than solving a specific algorithmic task, inclusion of environment within the computational representation, etc. Thus a deeper understanding of these inherent properties of BIST systems will provide new ideas for developing a interactive computing

To further appreciate this, consider for example interdependent societal infrastructure systems spanning large urban areas. They are the center of economic, commercial and social activities. The design of these urban areas, their rapid population growth, and sharing of the limited resources by their inhabitants has led to increased social interactions [47, 8]. Large scale information delivery, and access systems developed by today's computing companies such as Google, Yahoo, Akamai, etc. are examples of emerging socio-technical information infrastructure systems. Such regional and global scale infrastructure systems are spatially distributed, managed by different federal, state, and commercial entities and operate at multiple time scales. Despite this heterogeneity, based on certain basic economic and legal principles, these interdependent systems usually work seamlessly to provide uninterrupted services to the millions of individuals residing in the urban region. Under any reasonable definition, these are complex systems whose global behavior is a result of complicated interactions between constituent elements. For example, the spatial distribution of individuals in an urban region, their movement patterns, and their phone-calling patterns, all have a direct bearing on the structure and the design of wire-line and wireless telecommunication networks. A systematic understanding of such systems must therefore be able to represent the complex interdependencies between individual constituent elements and their dynamics. The focus is on understanding consequences of certain decisions or representing the interactions between individuals and the infrastructures rather than solving specific algorithmic question. The constituent BIST systems (e.g. transportation and urban populations) are tightly coupled and co-evolve: they are naturally viewed as large population ecologies. Computational models developed to represent these systems will necessarily have to clarify the role of interaction between constituent elements and the environment. This includes questions of what is being computed, the meaning and role of environment and acceptance of non-determinism as an elementary phenomenon.

1.2 Organization

The remainder of the chapter is organized as follows. Section 2 contains basic definitions and preliminary results. In Section 3, we discuss the theoretical foundations of interaction based simulation and modeling of BIST systems. Section 4 contains a discussion of the engineering principles necessary for design and implementation of large BIST system simulations. In Section 5 a practical operational system based on the theoretical and engineering foundations described in Section 3.1 to 4 is discussed. Finally, Section 6 contains concluding remarks and directions for future work.

2 Terminology and Preliminary Results

Informally, computer simulation is the art and science of using computers to calculate interactions and transactions among many separate algorithmic representations, each of which might be associated with identifiable “things” in the real world (at least in a world outside the simulation program). Because of the widespread use of computer simulations, it is difficult to give a precise definition of a computer simulation that is applicable to all the various settings where it is used. Nevertheless, it is clear that simulation has two essential aspects: dynamics generation and mimicry of the dynamics of another system by the dynamics of the simulation program. Thus we view simulations as comprised of the following: (i) a collection of entities with state values and local rules for state transitions, (ii) an interaction graph capturing the local dependency of an entity on its neighboring entities and (iii) an update sequence or schedule such that the causality in the system is represented by the composition of local mappings.

A **Sequential Dynamical System** (SDS) $\mathcal{S}$ over a given domain $\mathbb{D}$ of state values is a triple $(G, \mathcal{F}, \pi)$, whose components are as follows:

1. $G(V, E)$ is a finite undirected graph without multi-edges or self loops. G is referred to as the **underlying graph** of $\mathcal{S}$. We use n to denote $|V|$ and m to denote $|E|$. The nodes of G are numbered using the integers 1, 2, …, n.
2. For each node i of G, $\mathcal{F}$ specifies a **local transition function**, denoted by f_i. This function maps $\mathbb{D}^{\delta_i+1}$ into $\mathbb{D}$, where δ_i is the degree of node i. Letting $N(i)$ denote the set consisting of node i itself and its neighbors, each input of f_i corresponds to a member of $N(i)$.
3. Finally, π is a permutation of $\{1, 2, \ldots, n\}$ specifying the order in which nodes update their states using their local transition functions. Alternatively, π can be envisioned as a total order on the set of nodes.

A **configuration** $\mathcal{C}$ of $\mathcal{S}$ can be interchangeably regarded as an n-vector $(c_1, c_2, \ldots, c_n)$, where each $c_i \in \mathbb{D}$, $1 \leq i \leq n$, or as a function $\mathcal{C} : V \rightarrow \mathbb{D}$.

Computationally, each step of an SDS (i.e., the transition from one configuration to another), involves n substeps, where the nodes are processed in the *sequential* order specified by permutation π. The "processing" of a node consists of computing the value of the node's local transition function and changing its state to the computed value. The following pseudo-code shows the computations involved in one transition.

---

for $i = 1$ **to** n **do**

(i) Node $\pi(i)$ evaluates $f_{\pi(i)}$. (This computation uses the *current* values of the state of node $\pi(i)$ and those of the neighbors of node $\pi(i)$.) Let x denote the value computed.

(ii) Node $\pi(i)$ sets its state $s_{\pi(i)}$ to x.

end-for

We use $F_{\mathcal{S}}$ to denote the **global transition function** associated with $\mathcal{S}$. This function can be viewed either as a function that maps $\mathbb{D}^n$ into $\mathbb{D}^n$ or as a function that maps $\mathbb{D}^V$ into $\mathbb{D}^V$. $F_{\mathcal{S}}$ represents the transitions between configurations, and can therefore be considered as defining the dynamic behavior of SDS $\mathcal{S}$. A **fixed point** of an SDS $\mathcal{S}$ is a configuration $\mathcal{C}$ such that $F_{\mathcal{S}}(\mathcal{C}) = \mathcal{C}$.

The **phase space** $\mathcal{P}_{\mathcal{S}}$ of an SDS $\mathcal{S}$ is a directed graph defined as follows: There is a node in $\mathcal{P}_{\mathcal{S}}$ for each configuration of $\mathcal{S}$. There is a directed edge from a node representing configuration $\mathcal{C}$ to that representing configuration $\mathcal{C}'$ if $F_{\mathcal{S}}(\mathcal{C}) = \mathcal{C}'$.

It is possible to obtain restricted versions of SDSs by appropriately restricting the domain $\mathbb{D}$ and/or the local transition functions. We use the notation "(x,y)-SDS" to denote an SDS where 'x' specifies the restriction on the domain and 'y' specifies the restriction on the local transition functions. Thus for example, (BOOL, SYM)-SDS are SDS in which domain of state values is Boolean and each local transition function is symmetric. (BOOL, THRESH)-SDS are SDSs in which the domain of state values is Boolean and each local transition function is a simple-threshold function. And finally, (BOOL, NOR)-SDS are SDSs in which domain of state values is Boolean and each local transition function is the NOR function. **Synchronous Dynamical System** (SyDS), is a special kind of SDS, *without* node permutations. In a SyDS, during each time step, all the nodes *synchronously* compute and update their state values. Thus, SyDSs are similar to classical CA with the difference that the connectivity between cells is specified by an arbitrary graph. The restrictions on domain and local transition functions for SDSs are applicable to SyDSs as well.

Example 1. Consider a (BOOL, NOR)-SDS shown in Figure 1 (left). Let $\pi = (a, b, c)$. Each node a, b and c execute the local function $NOR(x, y, z)$. Phase space associated with the dynamical system when vertices are updated in the order a,b and c is shown in Figure 1 (right). Each tuple in the ellipse denotes the states of the nodes a, b and c in that order. Notice that the phase space does not have a fixed point. It turns out that SDS with NOR local functions can never have fixed points.

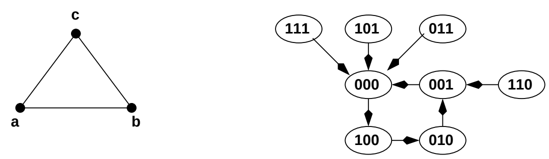

Fig. 1. Figure illustrating SDS and its phase space described in Example 1.

SDSs naturally capture the three essential elements of a computer simulation. The use of simple functions to represent each agent/entity is just an equivalent alternate representation of each individual as automata. The fact that each function depends locally on the state values of neighboring agents is intended to capture the intuition that individual objects comprising a real system usually have only local knowledge about the system. Finally, a permutation is an abstraction of the need to *explicitly* encode causal dependency.

The basic SDS model can easily be generalized in a number of ways including: (i) partial orders or schedules specified using formal languages, (ii) allowing stochastic local functions or interaction graphs, (iii) time varying SDS in which the topology or the local functions vary/evolve in time. These generalizations are important while modeling realistic BIST systems; see [7, 37, 54, 45, 52, 53] for additional details and examples.

Computational SDS (cSDS) arise naturally when each local function is viewed procedurally. cSDS are useful for formal specification, and analysis of infrastructure simulation systems and extend the algebraic theory of dynamical systems in two important ways. First, we pass from extremely general structural and analytical properties of composed local maps to issues of provable implementation of SDS in computing architectures and specification of interacting local symbolic procedures. This is related to successive reductions of cSDS to procedural primitives, which leads to a notion of cSDS-based distributed simulation compilers with provable simulated dynamics (e.g., for massively parallel or grid computation). Second, the aggregate behavior of iterated compositions of local maps that comprise an SDS can be understood as a (specific) simulated algorithm together with its associated and inherent computational complexity. We have called this the *algorithmic semantics* of an SDS (equivalently, the algorithmic semantics of a dynamical system or a simulation). It is particularly important to view a composed dynamical system as computing a specifiable algorithm with provable time and space performance.

2.1 SDSs as Elementary Models of Interactive Computation

The basic definition of SDS together with the above generalizations form an elementary model of interactive computation. The introductory chapter in this book identifies four distinguishing features of interactive computing, namely

- Computational Problem: A computational problem entails performing a task or providing a service, rather than algorithmically producing an answer to a question
- Observable Behavior: A computing component is now modeled not as a functional transformation from input to output, but rather in terms of an observable behavior consisting of interaction steps
- Environments: The world or environment of the computation is part of the model, playing an active part in the computation by dynamically supplying the computational system, or agent, with the inputs, and consuming the output values from the system. The environment cannot be assumed to be static, or even effectively computable; for example, it may include humans or other elements of the real world
- Concurrency: Computation is concurrent; the computing agent computes in parallel with its environment and with other agents that may be in it

SDS and its extensions adequately captures these four essential and distinguishing features and can be used to model practical BIST systems. The following example illustrates this point.

Example 2. **TRANSIMS** is a large-scale Federal Highway Administration (FHWA) funded transportation simulation project [9] that we co-developed over the last 10 years. In this project, an SDS-based approach was used to micro-simulate every vehicle in an urban transportation network (see [82] for an SDS specification). Each roadway is divided into discrete cells. Each cell is 7.5 meters long and one lane wide. Each cell contains either a vehicle (or a part of a vehicle) or is empty. The micro-simulation is carried out in discrete time steps with each step simulating one second of real traffic. In each time step, a vehicle on the network makes decisions such as accelerate, brake or change lanes, in response to the occupancy of the neighboring cells. We can represent the above model using the SDS framework. For ease of exposition, we assume a single lane circular road that can be modeled as a one dimensional array of cells. In this representation, each cell represents a 7.5 meter segment of the road. The variable gap is used to measure the number of empty cells between a car and the car ahead of it. In the following, let v denote the speed of the vehicles in number of cells per unit time, $v_{\max}$ denote the maximum speed and

$rand$ as a random number between 0 and 1. Finally, p_{noise} denotes the probability with which a vehicle is slowed by 1 unit. Each iteration consists of the following 3 sequential rules that are applied in parallel to all the cars:

1. Acceleration of free vehicles: **If** $v < v_{max}$, **Then** $v = v + 1$.
2. Braking due to cars in front: **If** $v > gap$, **Then** $v = gap$.
3. Stochastic Jitter: **If** $(v > 0)$ **AND** $(rand < p_{noise})$, **Then** $v = v - 1$.

To illustrate how an SDS based model can be constructed, let us consider a simple circular one lane road. One vehicle occupies one cell and has a given velocity. Let us assume that a vehicle can travel at one of three velocities: 0, 1 and 2. There are m vehicles and their initial positions are chosen at random. They are labeled 1 through m by the order in which they initially appear on the road. There is a schedule π that determines the update ordering. A vehicle at cell i with speed v is updated as shown in Table 1. This defines the local function at a node in the time evolving graph. Thus a vehicle at cell i with speed 1 that has two free cells ahead moves one cell ahead and gets the new speed of 2. At each time step t we can derive the associated dependency graph $G(t)$. The graph $G(t)$ has vertices $1, 2, \ldots, m$ corresponding to the vehicles. Two vehicles k and l are connected by an edge if the distance between them at time t is less than or equal to $v_{\max} = 2$. If the distance is larger they are independent by construction. (A vehicle only depends on what is ahead on the road.) Thus, for the configuration in Figure 2, we derive the dependency graph shown in Figure 2.

(Cell,Speed)	$i+1$ taken	$i+1$ free, $i+2$ taken	$i+1$, $i+2$ free
$(i,0)$	$(i,0)$	$(i,1)$	$(i,2)$
$(i,1)$	$(i,0)$	$(i+1,1)$	$(i+1,2)$
$(i,2)$	$(i,0)$	$(i+1,1)$	$(i+2,2)$

Table 1. The update rule for a single vehicle

Discussion

- The computational problem at hand is to represent traffic dynamics in a city. There is no explicit algorithmic description of this problem. Traffic is an *emergent* or *simulated* property. As discussed in [70, 76], traffic can be viewed as a chaotic system and thus even its simple properties are unlikely to be *predictable*.
- The description of the driver is not merely contained in the local rules, but is obtained via composing the time varying explicit interactions with other drivers. This notion of disaggregated normative agent is discussed further in Section 4.1. Moreover, this interaction is dynamic and the neighborhood changes all the time. In other words, the environment is not static. The driver interacts continually with the environment and co-evolves with it.

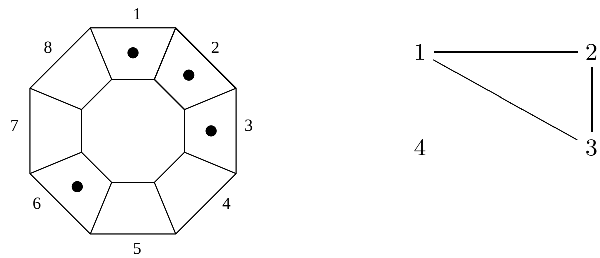

Fig. 2. A circular one-lane road divided into cells. A dot indicates that the given cell is occupied by a vehicle. The dependency graph $G(t = 0)$ associated to configuration to the left is shown to the right.

- The computation is inherently concurrent. The update order chosen is important. For instance, in the case of the single-lane system, updating the states from front to back acts like a perfect predictor and thus never yields clusters of vehicles. On the other hand, updating from back to front yields more realistic traffic dynamics [68, 70, 76].

The complete **TRANSIMS** system is described in Section 5 and models a number of other interesting features, including activity based traffic modeling, game theoretic behavior of individual travelers, co-evolution and effects of large scale transformational changes such as building new highways. The above example describes a simplified version of one of the **TRANSIMS** modules and is intended to convey the richness inherent in such systems. Nevertheless, the example drives the main point: SDSs and its extensions can serve as elementary models of interactive computation.

3 Theoretical Foundations

We describe an elementary theory of interaction based simulations abstracted as SDS. An elementary theory of simulation should yield theorems that are applicable to a class of simulations rather than to only particular members of this class. The first set of results outlined in Section 3.1 concern the structural properties of the interaction graph. The results are *independent* of the update order and the particular properties of the local functions. Section 3.2 outlines results that depend only on the properties of the local functions; they are independent of the interaction graph and the update order. Finally, in Section 3.3, we discuss results that pertain to all the three components of the definition.

3.1 Effect of BIST Network

Recently there has been a resurgence of research in complex networks, driven by a number of empirical and theoretical studies showing that network structure plays a crucial role in understanding the overall behavior of complex systems. See [23, 5, 2, 28, 33, 35, 34, 39, 71, 83] and the references therein for recent results in this active area. Another recent direction of research has been to determine random graph models that can generate such networks. Unfortunately, many of these random graph models, such as the preferential attachment model, are not suited for social network analysis.

Construction of BIST Networks Construction of BIST networks is challenging: in some cases data is easily available to construct the networks, while in the majority of other cases, although such data exists, it is not freely available. In yet other cases, the network has to be constructed by integrating a number of different databases. Finally, in case of social and ad-hoc networks, it is impossible at the current time to gather enough data to construct such networks. Thus simulation based tools are required for generating such networks. We describe two networks here: the social contact network and the mobile ad-hoc network. One is a social network, the other is formed by social interactions and the links are really a matter of convention, but nevertheless is best classified as a infrastructure network. Important examples of other BIST networks that have to be constructed by integrating various information sources and simulations include the route level IP network, the Gene annotation networks and Protein-Protein Interaction networks.

Example 3. Consider a social network that captures the interaction between individuals moving through an urban region [33, 7]. This information can be abstractly represented by a (vertex and edge) labeled bipartite graph G_{PL}, where P is the set of people and L is the set of locations. If a person $p \in P$ visits a location $l \in L$, there is an edge $(p, \ell, label) \in E(G_{PL})$ between them, where $label$ is a record of the type of activity of the visit and its start and end points. Each vertex (person or location) can also have labels. A person's various labels correspond to his/her demographic attributes such as age, income, etc. The labels attached to locations specify the location's attributes such as its x and y coordinates, the type of activity performed, maximum capacity, etc. Note that there can be multiple edges between a person and a location recording different visits. Figure 3 shows an example of a bipartite graph. Part (a) of Figure 3 shows an example of a bipartite people-location graph G_{PL} with two types of vertex representing four people (P) denoted by filled circles and four locations (L), denoted by squares. Figure 3 parts (b) & (c), show two distinct projections of the basic network that can be defined and constructed from this information. The graphs G_P and G_L induced by G_{PL}. G_P is the temporal people-people-spatial-proximity graph. It connects two individuals by edges if they were in spatial proximity during some time of the day. G_L is the building-building temporal graph. Two buildings are joined by an edge in a time period if an individual left one of the buildings in that period and arrived at the other building in the same time-period. Figure 3 part (d) shows the static projections of G_P^S and G_L^S resulting from ignoring time labels.

We point out that simulations appear to be the only way to construct such networks. Contrast this with the electrical grid: although it might be hard to obtain the data, the data certainly exists with government agencies and private companies.

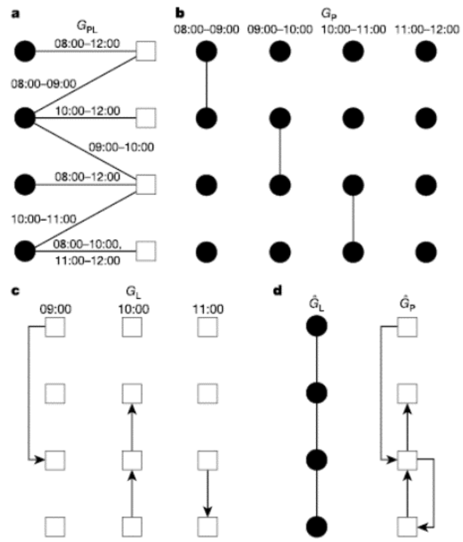

Fig. 3. Figure depicting a social contact network described in Example 3. (a) shows the bipartite graph G_{PL}. (b) and (c) show two distinct temporal projections of G_{PL}, namely G_P and G_L and (d) shows the static projections G_P^S and G_L^S resulting from ignoring time labels.

Example 4. A synthetic vehicular adhoc telecommunication network is obtained by assigning one or more wireless devices to drivers, vehicles and other individuals in an urban region. Each vertex in the adhoc telecommunication network corresponds to a transceiver and two nodes are joined by an edge if and only if they are within each other's radio range. Note that to construct such a network, one needs the following: a detailed time varying location of transceivers, information on the characteristics of the transceiver and time varying activity related to the transceiver (on and off patterns). Again, as in the case of social contact networks, it is hard to get data for such networks and simulation based data integration and creation methods appear to be necessary. We used the section of downtown Portland, Oregon, shown in Figure 4 for illustration. More details on the structural properties of realistic vehicular ad-hoc networks can be found in [13, 14, 25].

Important Notes:

- Notice how various components of network constructions played a role in the above examples. In Example 3, the underlying population and the infrastructure remained invariant. We simply varied the interaction criteria. In Example 4, the synthetic individuals had to be endowed with additional attributes such as a mobile wireless device. The interaction criteria is different and is defined with respect to the wireless device and is in this case the radio range of the individual transceivers (transmitter and a receiver).
- The two networks have differing levels of fidelity in terms of temporal evolution. In Example 3, if the intended application is disease propagation, then time

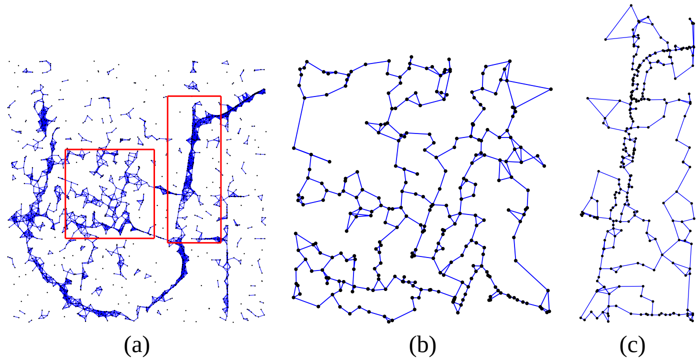

Fig. 4. Versions of an adhoc telecommunication network formed by assigning transceivers to individuals in cars on a section of Portland road network discussed in Example 4. (a): Network topology when all the transceivers were assigned the same power. (b) and (c) show parts of the network when power control algorithms in [58] were applied to reduce the overall interference.

scales could be relatively large, on the order of minutes to hours. In contrast, the telecommunication adhoc network formed needs to be represented and computed at extremely small time scales (milli-seconds), since loss in radio range implies loss in data packets. Notice that as society becomes ever more digital, social networks can more appropriately be defined not only over individuals but also over digital devices capable of handling specific tasks.

- While we have not elaborated it here, individual transceivers can choose to send messages to other specific transceivers (e.g. text messages on a phone): this yields yet another social network with communication devices as nodes and an edge between two devices when they send a message to each other. Such a network rides on the top of the rapidly evolving communication network that is described here.

Measurement and Analysis of BIST networks Once a complex network is constructed, we study the following interrelated questions: (i) discovering new measures that provide information about the network's structure and dynamics (ii) fast and provable algorithms for computing network measures over very large social and infrastructure networks. Some important observations based on results in [14, 33, 34, 35] include: (i) Social and infrastructure networks are not necessarily scale free or small world networks [33, 34, 35], (ii) Structural measures for real infrastructure and social networks are often different from similar measures for classical random networks, and (iii) Social networks have high local clustering. In contrast, many physical networks such as power and transport networks have very low clustering coefficient.

We illustrate the range of static analysis by describing important structural results pertaining to social contact networks such as the ones described in Example 3. See [33, 34, 35] for a more comprehensive discussion on this subject. In the bipartite

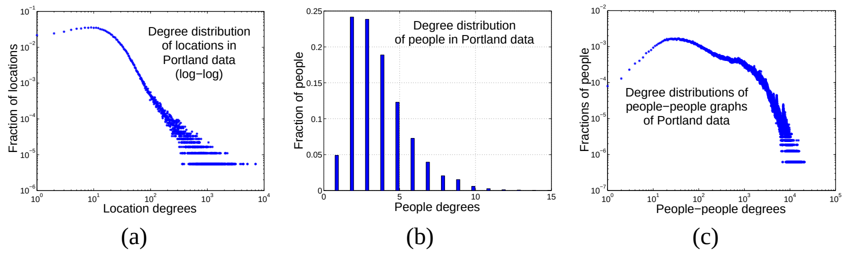

Fig. 5. (a) and (b) Degree distributions of locations and people in the bipartite graph G_{PL} for Portland data. The location degrees range from 1 to 7091, people degrees range from 1 to 15. (c) Degree distribution of people-people graph projection obtained from the original bipartite graph.

graph G_{PL} for the city of Portland, there are 1615860 (1.6 million) individuals, 181230 (181K) locations, and 6060679 (6.1 million) edges. Figure 5 (a) and (b) shows the degree distributions of the locations and people in the bipartite graph, G_{PL} for the Portland data. Note that a large part of the degree sequence of locations follows a power-law distribution, i.e., $n_k \propto k^{-\beta}$, where n_k denotes the number of locations of degree k; for the Portland data, $\beta \approx 2.8$. The degree distribution of people is roughly Poisson. The degree sequence of people in the people-people graph G_P is shown in Figure 5 (c) and looks quite different than the degree sequence of G_{PL}. The graph G_P for Portland is not fully connected, but has a giant component with 1615813 people. The clustering coefficient p of G_P: it is about 0.57 which is substantially higher than clustering coefficients for infrastructure networks.

Next, we describe two structural measures that provide further evidence into how *well connected* today's urban social networks are. First, consider graph expansion. We consider the two standard notions of expansion in the graph G_P. The edge expansion of a subset $S \subseteq P$ is defined as the ratio

$$\frac{|\{e = (u, v) :\ (u, v) \text{ is an edge and } u \in S, v \notin S\}|}{|S|}.$$

The vertex expansion of a subset $S \subseteq P$ is defined as the ratio $|\{u \notin S : (u, v) \text{ is an edge and } v \in S\}|/|S|$. The edge (vertex, respectively) expansion of G_P is the minimum, taken over all $S \subset P, |S| \leq |P|/2$, of the edge (vertex, respectively) expansion of S. The vertex and edge expansions are important graph-theoretic properties that capture fault-tolerance, speed of data dissemination in the network, etc. Roughly, the higher the expansion, the quicker the spread of any phenomena (disease, gossip, data etc.) along the links of the network. Random sampling based estimates of vertex and edge expansion are shown in Figure 6. The Y-axis plots the smallest expansion value found among the $500,000$ independent samples; the X-axis plots the set size S as a percentage of the total number of vertices in the graph (the sampling probability). The plots labeled "Vertex expansion-2" and "Edge expansion-2" in Figure 6 show the expansion in the graph G_P, while the plots marked

"Vertex expansion-1" and "Edge expansion-1" show the same quantity on a sparser people-people graph - the graph is made sparser by only retaining edges between individuals who came in contact for at least one hour. The graphs make two points: (i) as expected expansion becomes smaller as the contact graph gets sparser, and (ii) even for sparse contact networks the expansion values are quite high.

Another important structural measure (informally called *shattering*) is to determine the ability to disconnect a social or an infrastructure network by removing high connectivity nodes. Figure 6(b) and (c) show these plots for 3 infrastructure networks and urban social networks respectively. Notice the remarkable difference between the plots: they show that while infrastructure networks are prone to targeted failures, social networks are very robust. Targeted failures correspond to removal of high degree nodes. For social networks, this corresponds to removing individuals by quarantining or vaccinating them in case of epidemics, with large number of social contacts. This connectivity property of the social network turns out to be the *Achilles heel*: while strong connectivity is important for the day-to-day functioning of the social system, it is a weakness in controlling the spread of infectious diseases. In other words, The high expansion and inability to shatter social networks implies that contagious diseases would spread very fast, and making early detection imperative to control disease.

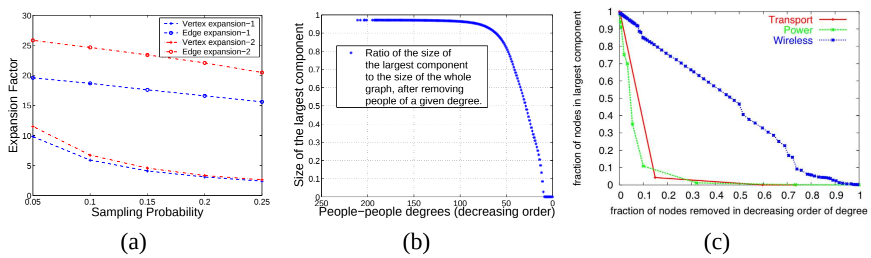

Fig. 6. (a) Expansion of the people-people graph: the plots marked "Vertex expansion-2" and "Edge expansion-2" show the vertex and edge expansion for the graph G_P, while "Vertex expansion-1" and "Edge expansion-1" show the corresponding quantities in the graph obtained by retaining only those edges that involve an interaction of at least 1 hour. This leads to a much sparser graph and correspondingly lower values of vertex and edge expansions. (b) Plots showing the relative ease with which we can break infrastructure networks by removing nodes of high connectivity. (c) In contrast to (b), figure (c) shows that urban social networks are very hard to shatter.

3.2 Effect of Local Functions

In this section, we give examples of results that depend solely on the properties of the local functions. We give three examples and restrict ourselves to local functions with Boolean domains; see [19, 15, 21, 51]. Given an SDS $\mathcal{S}$ over a domain $\mathbb{D}$, two configurations $\mathcal{I}$, $\mathcal{B}$, and a positive integer t, the t-REACHABILITY problem is to decide whether $\mathcal{S}$ starting in configuration $\mathcal{I}$ will reach configuration $\mathcal{B}$ in t or fewer time steps. We assume that t is specified in binary. (If t is specified in unary, it is easy to solve this problem in polynomial time since we can execute $\mathcal{S}$ for t steps and check whether configuration B is reached at some step.) Given an SDS $\mathcal{S}$ over a domain $\mathbb{D}$ and two configurations $\mathcal{I}$, $\mathcal{B}$, the REACHABILITY problem is to decide whether $\mathcal{S}$ starting in configuration $\mathcal{I}$ ever reaches the configuration $\mathcal{B}$. (Note that, for $t \geq |\mathbb{D}|^n$, t-REACHABILITY is equivalent to REACHABILITY.) Given an SDS $\mathcal{S}$ over a domain $\mathbb{D}$ and a configuration $\mathcal{I}$, the FIXED POINT REACHABILITY problem is to decide whether $\mathcal{S}$ starting in configuration $\mathcal{I}$ reaches a fixed point.

1. The REACHABILITY and t-REACHABILITY problems are solvable in polynomial time for (BOOL, NOR)-SDSs for which the number of independent sets in the underlying graph is polynomial. For any (BOOL, NOR)-SDS, every transient in the phase space is of length 1 and the phase space does not have fixed points.
2. Given an n-node (FINRING, LINEAR)-SDS $\mathcal{S}$ over a finite domain $\mathbb{D}$, the FIXED POINT REACHABILITY problem for $\mathcal{S}$ can be solved using a number of algebraic operations that is polynomial in n and $|\mathbb{D}|$. When the domain $\mathbb{D}$ is Boolean and the operators of the unitary semi-ring are OR (+) and AND (*), each linear local transition function is either XOR (exclusive or) or XNOR (the complement of exclusive or). Thus, the FIXED POINT REACHABILITY problem for such SDSs can be solved efficiently.
3. Let $\mathcal{S} = (G, \mathcal{F}, \pi)$ be a (BOOL, THRESH)-SDS whose underlying graph G has n nodes and m edges. From any initial configuration $\mathcal{I}$, $\mathcal{S}$ reaches a fixed point after at most $\lfloor (m+n+1)/2 \rfloor$ steps. Thus, t-REACHABILITY, REACHABILITY and FIXED POINT REACHABILITY problems for (BOOL, THRESH)-SDSs can be solved in polynomial time.

3.3 Composite Analysis of SDS

Finally, we consider examples of composite analysis of SDS. Following [17], we say that a system is predictable if basic phase space properties such as reachability and fixed point reachability can be determined in time which is polynomial in the size of the system specification. It can be shown that very simple SDSs are computationally universal for the appropriate space/time complexity class (see [15, 21]). For example there exist constants d_2, p_2 and n_2 such that the t-REACHABILITY, REACHABILITY and FIXED POINT REACHABILITY problems for (BOOL, SYM)-SDSs are **PSPACE**-hard, even when all of the following restrictions hold: (a) The maximum node degree in the underlying graph is bounded by d_2. (b) The pathwidth (and hence the treewidth) of the underlying graph is bounded by p_2. (c) The number of distinct local transition functions used is bounded by n_2.

Due to the particular proof technique used, these results naturally extend to yield general computational universality. For instance, we show that the reachability problem for *very simple* SDS (e.g. SDS in which the domain of state values is Boolean and each node computes the same symmetric Boolean function) is **PSPACE**-hard:

this implies that the systems are not easily predictable. In fact, the results imply that no prediction method is likely to be more efficient than running the simulation itself. By allowing an exponential memory at each node or allowing exponentially many nodes, one can obtain **EXPSPACE**-hardness results. An important implication of this (stated informally) is the following: *the optimal computational strategies for determining the structural properties of such complex dynamical systems are interaction based simulations*. Moreover the systems for which the hardness results hold are so simple (essentially, local transition functions can be simple threshold or inverted thresholds) that any realistic socio-technical system is likely to have such systems embedded in them. See [17, 66, 40, 85] for additional discussion on this topic.

As another illustration of the general complexity theoretic results that can be obtained as regards to SDSs, we consider the predecessor existence problem. Given an SDS $\mathcal{S}$ and a configuration $\mathcal{C}$, the PREDECESSOR EXISTENCE (or PRE) problem (a.k.a pre-image existence problem) is to determine whether there is a configuration $\mathcal{C}'$ such that $\mathcal{S}$ has a transition from $\mathcal{C}'$ to $\mathcal{C}$. Apart from the decision version, we also consider the problems of counting the number of predecessors (the counting version, denoted by #-PREDECESSOR EXISTENCE), deciding if there is a unique predecessor (the unique version, denoted by UNIQUE-PREDECESSOR EXISTENCE) and if there are two predecessors of the given configuration (the ambiguous version, denoted by AMBIGUOUS-PREDECESSOR EXISTENCE). Using the concept of *simultaneous local reductions*, it is possible to obtain results that simultaneously characterize the complexity of the PREDECESSOR EXISTENCE, #-PREDECESSOR EXISTENCE, UNIQUE-PREDECESSOR EXISTENCE and AMBIGUOUS-PREDECESSOR EXISTENCE problems for SDS and SyDS. The results are summarized in Figure 7 and are proved in [20]. These are local transformations that simultaneously yield the hardness for decision, counting, unique and ambiguous versions of the problem. Such a reduction allows us to tightly relate the computational complexity of these problems; see [30, 49, 50] for more discussion on simultaneous local reductions. The easiness results are obtained using generic algorithms that exploit the underlying structure of the interaction graph and the semantics of the local transition functions. The algorithms are generic in the sense that the same basic algorithm can be used to compute solution to the decision, counting, ambiguous and unique versions of the problem by merely supplying the appropriate semantics for the semi-ring operations that are carried out; see [80].

3.4 Formal Specifications and Local Simulation Compliers

Discrete dynamical systems are a natural mathematical language for formally specifying large scale interacting systems. Recently SDS and abstract state machines (ASM)[2] have been used for formally specifying the several modules of the telecommunication system [24, 59]. Ideally, we would like to express the BIST systems using *higher level SDSs*, i.e. SDSs with more expressive local functions and interaction networks. In contrast, *simpler* SDSs, i.e. SDSs with less expressive local functions

[2] See `http://www.eecs.umich.edu/gasm/`.

The PRE problem is **NP**-complete for the following restricted classes of SDSs. In most cases, the #-PREDECESSOR EXISTENCE problem is #**P**-complete, the AMBIGUOUS-PREDECESSOR EXISTENCE problem is **NP**-complete and UNIQUE-PREDECESSOR EXISTENCE problem is $\mathbf{D^P}$-complete (using randomized reductions).

1. **Identical and/or restricted class of functions:**
   a) (BOOL, THRESH)-SDSs where each node computes the same k-simple-threshold function for any $k \geq 2$,
   b) (BOOL, TALLY)-SDSs in which each node computes the same k-tally function for any $k \geq 1$,
2. **Restricted Graphs**:
   a) SDSs over the Boolean domain where at most one local transition function is not symmetric and the underlying graph is a *star*,
   b) SDSs over the Boolean domain and the underlying graph is a *grid*,
   c) (BOOL, SYM)-SDSs whose underlying graphs are *planar*.

The PRE problem is in **P** for the following classes of SDSs.

1. for (FIELD, LINEAR)-SDSs, (BOOL, AND)-SDSs and (BOOL, OR)-SDSs with no restrictions on the underlying graph,
2. for (BOOL, SYM)-SDSs when underlying graphs have bounded treewidth,
3. for SDSs when underlying graph is simultaneously bounded degree and and bounded treewidth with no restriction on the local transition functions (other than that the functions are over finite domain).

Fig. 7. Example of complexity theoretic results that can be proven for special classes of SDS. Note the inter-play between the graph structure and function complexity. Although the results are shown only for PRE problem and its variants, it is possible to obtain similar results for other problems such as garden of eden states, etc. These results also imply analogous results for Discrete Hopfield networks, concurrent transition systems and other related models.

and regular interaction networks are likely to be more suitable for finding efficient mappings of the SDSs on HPC architectures. This is because the language (model) that is most convenient to describe the underlying system might not necessarily be the best model for actual simulation of the system on a HPC architecture. Thus it is conceivable that such simpler systems obtained via translation could be mapped on HPC architectures and the resulting maps could be analyzed for performance bottlenecks. Simpler systems can potentially also be used to verify the correctness of the ensuing protocols. To achieve this, such translations should be efficient and preserve the basic properties across the original and the translated system. The constructions given as part of the simulation results in [19, 21] can be viewed as *local simulation compilers* that transform one type of SDS to a simpler kind of SDS in such a way that (i) the translation is local and efficient and (ii) relevant features of the phase space of the original SDS are captured appropriately in the phase space of the simpler SDS. In recent years (see [40, 85, 42, 63] and the references therein), several authors have suggested building *cellular automata based computers* for simulating physics. We believe that *SDS based computers* are better suited for simulating BIST systems. In

[62], Margolus proposes a DRAM based architecture for large scale spatial lattice computations, also see DeHon [32]. Simulation compilers as discussed above will form the basis for implementing Simfrastructure like simulations on massively parallel architectures such as FPGAs. See [82] for a recent study.

3.5 Implications for Other Computational Models

The complexity theoretic results for SDS can be used to yield lower (and upper) bounds on the complexity of reachability problems for other computational models of discrete dynamical systems. These include:

1. Classical Cellular Automata (CA), (see for example, [85]) systolic arrays proposed by Kung et al. [56] and *graph automata* [72], which are a widely studied class of dynamical systems in physics and complex systems and have been used in massively parallel computing architectures.
2. Concurrent transition systems (CTS) have been widely studied as formal models of concurrent processes. They have been used to specify communication protocols and concurrent programs in the context of distributed computing.
3. Discrete *recurrent Hopfield networks* [36, 73, 73] which are used in machine learning and image processing.

The results can be used to characterizations of the complexity of state executability problems for CTSs, discrete Hopfield networks and cellular automata in terms of (i) the power of individual automata, (ii) the size of the alphabet for encoding messages, (iii) the inter-connection topology and (iv) the method of communication (e.g. channels, action symbols).

4 Engineering BIST Systems

An important factor in building simulations of BIST systems is the size and scope of the systems that need to be represented. For example, infrastructure simulations should be able to represent over 10^6 entities and cover large geographical areas, the size of medium sized metropolis. A telecommunication simulation system representing a medium sized city should be able to represent 10^9 transceivers and 10^{12} packets per hour. As a result, building such systems requires new engineering principles for a high resolution HPC oriented representation. Classical methods for representing agents and their interactions will not scale beyond a certain point. Another interesting problem involves methods related to spatio-temporal data collection, integration and validation. Building such simulations involves, on the one hand, integrating large numbers of databases, streaming datasets and results from earlier simulation runs in a consistent manner and on the other hand, developing efficient methods for storing and analyzing data that is produced by such simulations. We discuss two interrelated topics below.

4.1 Concept of Agency: A Disaggregated Interactive, Normative Representation

Another issue to consider while implementing large simulations is that of agent encapsulation. In the past, most work on agent-based simulations has been implemented using object oriented computing languages and as a result people have a found natural one-to-one mapping of agents onto objects. This simplifies the task of debugging and implementing the agent based simulation architecture. Unfortunately, this approach does not scale while implementing large BIST systems. The notion of agency is much more abstract than usually studied in the literature and is based on the notion of composition and interaction. By composition, we mean that the functionality associated with an agent is obtained by composing (both structurally and functionally) its various incarnations or avatars. By interaction, we mean that a specific functionality of an agent depends on the behavior of other agents interacting with it. For instance, in the traffic simulation (**TRANSIMS**), an agent is sometimes a driver and sometimes a parent and sometimes an office worker. When assuming the role of a driver, the agent's speed is not only dependent on his own rules but the speed of other drivers around him. The SDS based view again provides a natural mathematical framework to represent this notion of agency.

PARameterized Approximate Local and Efficient aLgorithms (**PARALEL**) provide a way to address the scaling issue. As discussed above, in simulating large systems with tens of thousands (or more) of interacting elements, it is computationally infeasible to explicitly represent each entity in detail using, perhaps, naive one agent-one encapsulated software object representational ideas. A common method of simulating such systems is to use parameterized representations of entities. The goal is to capture different behaviors of the system using different sets of parameters. The concept corresponds to having a normative representation of each abstract agent. A parameterized representation allows efficient use of computational resources. Indeed, even in systems with only tens of thousands of entities, the set of potential interactions among the entities is so large that parameterized representations are desirable, if not absolutely necessary to simulate the interactions in an efficient manner. The basic ideas behind agent abstraction are found in the concept of **PARALEL** algorithms:

- **PAR**ameterized, in that a single basic algorithm with a correct set of input parameters is capable of representing a class of algorithms,
- **A**pproximate, in that their behavior closely approximates an exact algorithm achieving a given task,
- **L**ocal, in that the information required by such algorithms is local as opposed to global, and
- **E**fficient, in that they are very fast and can be executed efficiently on both sequential and distributed shared memory multiprocessor architectures a-**L**-gorithms.

The concept of local algorithms is akin to the recently independently introduced concept of decentralized algorithms [55] and also to the classical concept of distributed algorithms. The approximate behavior is also pertinent at two levels. At the basic level an approximate algorithm closely models the behavior of each physical entity. At a global level, an approximate solution implies that the composed local algorithms representing each agent along with the update mechanism approximate the global system dynamics. The global level of approximation is more important, although the local level cannot be completely ignored.

Example 5. **Normative Drivers in Traffic Simulations.** Consider the rules for a driver update given in Example 2 In spite of their simplicity, these rules produce fairly realistic traffic flow characteristics and can in the limit, approach the fluid dynamics models studied in traffic flow theory [68, 70, 76]. The traffic pattern evolution as a function of the density $\rho = m/n$ (m is the number of cars in a given a period of time on a road segment of length n measured in number of cells) exhibits a threshold value for congestion. Figure 8 shows illustration of traffic flow characteristics produced by the above set of rules for a one-lane road with periodic boundary conditions. See [69] for additional discussion.

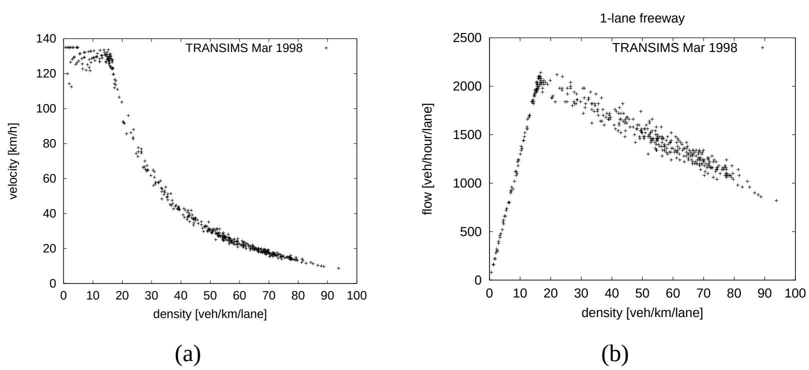

Fig. 8. Figures representing various traffic flow characteristics.

4.2 Efficient Storage and Regeneration

The simulations of BIST systems described here produce extremely large quantities of data. For example, simulating an adhoc packet switched network with a million moving transceivers for even 15 minutes produces time varying network requiring gigabytes of memory and packet level data requiring terabytes of memory. It is therefore impossible to exhaustively store the data generated while running these simulations. This motivates the need for computationally efficient data storage and methods with the following requirements: (i) efficiency in terms of space and time complexity and in many cases capability to run in an online setting, and (ii) the stored data should have enough information to allow recreation of certain dynamic features observed while running the simulations. We can equivalently view this as a semantic compression step.

The next step is efficient (re)-generation of data (including networks). Generation of random graphs and random data sets allow us to test scalability as well as the semantic properties of simulations. Re-generated data is necessary to recreate data that could not be stored while running the larger simulations. Re-generation methods can be viewed as reduced simulations; they allow one to generate certain dynamics of interest without resorting to expensive runs of the large simulation. For example, in [12] a system is described to store and regenerate statistically equivalent packet streams arriving at their destination succinctly using signal theoretic and statistical methods. The size of the stored model is much smaller than the original data. The regeneration step uses the Markov Chain Monte-Carlo method. The regenerated packet sequences are statistically indistinguishable from the original packet sequence when compared using basic quality of service measures such as throughput, jitter, skips, repeats, etc. The methods appear to yield compression ratios of over 100,000 while being able to recover many of the measures within 1% error. Similar methods can be devised to store and regenerate large BIST networks. The compression methods store structural properties of the network. The regeneration methods then use stochastic methods to re-generate the graphs. The random graphs so generated are ”similar” to the original networks and can be constructed in a fraction of the time required to construct original networks.

5 A Practical Interaction Based System: Modeling Interdependent Urban Infrastructures

As an example of the theoretical framework described in the preceding sections, we will describe **Simfrastructure**: a high-performance service oriented agent based modeling and simulation system for representing and analyzing interdependent infrastructures. See [4, 26, 27, 29, 44, 57, 61, 31, 75, 86] and additional references in the following sections for other examples of similar efforts. **Simfrastructure** can represent and analyze interdependent urban infrastructures including transportation, telecommunication, public health, energy, financial (commodity markets)[3]. In conjunction with a representation of the urban population dynamics and the details of the built infrastructure, such modeling systems can be viewed as *functioning virtual cities*. A unique feature of tools such as **Simfrastructure** is their ability to represent entire urban populations at the level of individuals, including their activities, movements and locations. The ability to generate an urban population, move each person on a second-by-second basis, and monitor the individual’s interaction with others and the physical infrastructures enables the understanding of infrastructure operations and interdependencies at an extreme but practical level of detail.

[3] See `http://ndssl.vbi.vt.edu/` for more details.

A connected collection of such urban infrastructure simulations allow analysis of urban infrastructure interdependencies through integrated functional data flow architectures. In brief, this functionality derives from population-mobility data generated by the simulation and modeling framework for the transportation sector. The simulation produces a synthetic population with demographics assigned to every individual. We track the second-by-second activities and locations of each individual by tying population information to detailed maps of urban infrastructures. This information drives each of the infrastructure simulations and is shared among the various infrastructure sector modules through a common interface. This also allows us to provide feedback between modules regarding infrastructure changes that arise in one sector during the course of a simulation and are likely to affect the behavior of other infrastructures. With the ability to simulate multiple infrastructures and their interdependencies in large urban regions, these systems provide planners and decision makers with an unprecedented modeling and analysis capability. Figure 9 shows a schematic view of the interdependent infrastructure simulation architecture.

5.1 A Service Oriented Architecture of Simfrastructure

We have recently completed a design and initial prototype implementation of **Simfrastructure** using web services based globally scalable architecture. The new design of the system specifically aims to scale Simfrastructure to represent entire countries and over time entire global populations. The only way to achieve such unprecedented scalability is to use web services architecture combined with Grid Computing infrastructure. We have recently demonstrated the design by constructing extremely detailed proto-populations of individuals residing in states along the US Eastern seaboard consisting of approximately 100 Million individuals. This architecture takes care of ensuring that the simulations have the data that they need to operate, allow direct discovery of available services, and facilitate the integration of new services. The system design allows simulation modules to be run on any available computation resource in a way that is transparent to the user. The use of existing web services standards, allows any architecture or programming language to be supported.

The newly developed architecture makes it easy for organizations to add their own simulations and analysis tools into the system. One novel aspect of the architecture is the ability for different organizations to host the same simulation applied to different geographic areas. These instances will be able to communicate through web services to collaborate on a larger problem. For instance, a transportation system simulation could be run at each Metropolitan Planning Organization (MPO) covering the local urban region. The simulations running at each MPO could then exchange the traffic exiting each local area and entering an adjacent area. This exchange could be expanded to include bus, rail, and air traffic to aid in epidemiological modeling at the national level. Note that the system formed in this way is not predetermined, but is self-organized based on the currently available services.

The architecture also allows the implementation of a particular service to be easily updated or replaced without affecting current users of the service. Multiple providers of a service can co-exist, each with a different trade-off (e.g., resolution vs. execution time). The request for a service will be decoupled from the execution of the service so that a user simply makes a request that a service be performed. Attached to the request are conditions that must be met such as monetary cost, completion time, security requirement, etc. These requests need not be computational, but may be for services provide by other individuals or organizations. Software brokers examine these requests and match them to available resources.

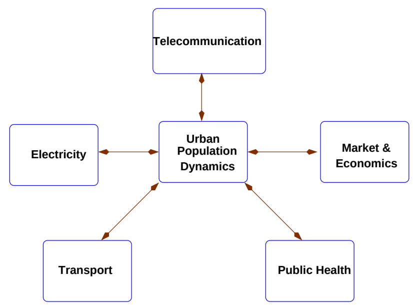

Fig. 9. A schematic diagram **Simfrastructure**: an interdependent urban infrastructure simulation and modeling framework.

Currently, **Simfrastructure** has working models for the following infrastructures: (i) Synthetic populations and urban environments, (ii) transportation, (iii) commodity markets, (iv) integrated telecommunication, (v) public health, (vi) electrical power. Below we describe each of these modules briefly. We will end the section with illustrative use cases.

5.2 Synthetic Protopopulations and Urban Environment Representation

A detailed population mobility and the associated built urban infrastructure is the central piece of such simulations. It provides a common interface for the flow of information between all the infrastructure sector simulations. All information describing the synthetic population and elements of the built urban environment resides in this module. In addition, changes in the urban infrastructure that arise during the course of a simulation and constrain activities and locations of the population pass between the modules through this module, where sector-specific information is transformed into a common format. The module makes information available to the other infrastructure simulations in the form of a consistent data structure, called protopopulations: they are synthetic populations whose resolution, fidelity and quality can be varied depending on the nature of the application.

A protopopulation is a collection of synthetic people, each associated with demographic variables drawn from any of the demographics available and extracted from the census [16, 77, 78]. Protopopulations can represent a person, a vehicle, or an infrastructure element such as a hospital or a switch. Here, for illustration, we will concentrate on creation of synthetic urban populations. Figure 10 shows a schematic diagram. Joint demographic distributions can be reconstructed from marginal distributions available in typical census data using an iterative proportional fitting (IPF) technique. Each synthetic individual is placed in a household with other synthetic people and each household is placed geographically in such a way that a census of the synthetic population is statistically indistinguishable from the original census, if aggregated to the block group level. Synthetic populations are thus statistically indistinguishable from the census data. Since they are synthetic, the privacy of individuals within the population is protected. The *synthetic individuals* carry with them a complete range of demographic attributes collected from the census data, including variables such as income level and age. Next, a set of activity templates for households is created, based on several thousand responses to an activity or time-use survey. These activity templates include the types of activities each household member performs and the time of day they are performed.

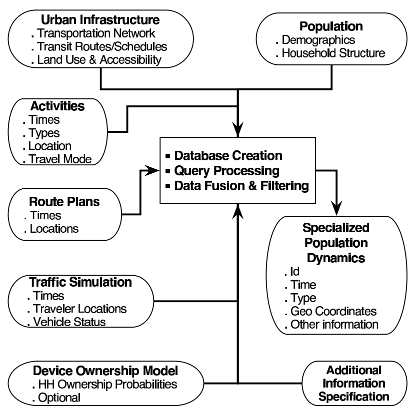

Fig. 10. Schematic diagram showing how various databases are integrated to create a synthetic population.

Each synthetic household is then matched with one of the survey households, using a decision tree based on demographics such as the number of workers in the household, number of children of various ages, etc. Next, the synthetic household is assigned the activity template of its matching survey household. For each household and each activity performed by this household, a preliminary assignment of a location is made based on observed land-use patterns, tax data, etc. This assignment must be calibrated against observed travel-time distributions. However, the travel-times corresponding to any particular assignment of activities to locations cannot be determined analytically. Indeed, the urban transportation system is a canonical example of complex system wherein global behavior arises from simple local interactions. Using techniques from combinatorial optimization, machine learning and agent based modeling we then refine the population, their activity locations and their itineraries [9].

The time varying, spatially placed, synthetic population constructed in the above manner can be enhanced for other uses. For instance, we used data fusion techniques to assign these individuals: telecommunication devices (cell phones, pagers, etc.), time varying demand for electricity, water and other such commodities. Note that such data is impossible to collect and can only be created using methods described here.

This produces synthetic individuals that just like real individuals can now call other individuals, consume various resources during the day and carry out other activities like eating, socializing, shopping, etc. An important point to note here is that such data is impossible to collect by mere measurements or surveys: it is the output of the agent based models such as the ones developed in [9].

5.3 Transportation Sector

Large scale microscopic simulation of transportation systems has become possible over the last few years. See [31, 75, 9] for examples of efforts in this regard. A prototypical question that can be studied with such simulations is the economic and social impact of building a new freeway in a large metropolitan area. Systems such as **TRANSIMS** conceptually decompose the transportation planning task into three time scales.

First, a large time-scale associated with land use and demographic distribution as a characterization of travelers. In this phase, demographic information is used to create *activities* for travelers. Activity information typically consists of requests that travelers be at a certain location at a specified time. They include information on travel modes available to the traveler. A synthetic population is endowed with demographics matching the joint distributions given in census data. Observations are made on the daily activity patterns of several thousand households (from survey data). These patterns are used as templates and associated with synthetic households with similar demographics. The locations at which activities are carried out are estimated while taking into account observed land use patterns, travel times, and dollar costs of transportation. Second, an intermediate time-scale consists of planning routes and trip-chains to satisfy the activity requests. This module finds minimum cost paths through the transportation infrastructure consistent with constraints on mode choice. An example constraint might be: “walk to a transit stop, take transit to work using no more than 2 transfers and no more than 1 bus” [9]. Finally, a very short time-scale is associated with the actual execution of trip plans in the network. This is done by a simulation that moves cellular automata corresponding to the travelers through a very detailed representation of the urban transportation network [68] . Examples 2 and 5 have already discussed some of these aspects. The simulation resolves traffic down to 7.5 meters and times down to 1 second. It provides an updated estimate of link costs, including the effects of congestion, to the Router and location estimation algorithms, which produce new plans. This feedback process continues iteratively until convergence to a steady state in which no one can find a better path in the context of everyone else’s decisions. The resulting traffic patterns are matched to observed traffic.

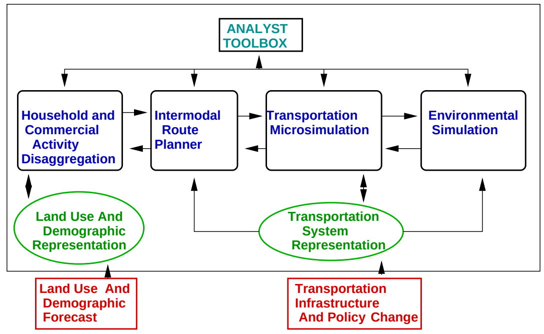

Fig. 11. Data flow in the **TRANSIMS** simulation system, proceeding from left to right. Input data comes from the U.S. census and metropolitan planning organizations. We generate a synthetic population whose demographics match the census; give each household an appropriate set of activities; plan routes through the network; and estimate the resulting travel times. The dotted lines represent feedback pathways, along which data flows from right to left, in the system.

A substantial effort has been spent on calibration and validation of the output produced by **TRANSIMS**; see [9, 68] for details. First, the design of the system is based on SDS. Second, various microscopic and macroscopic quantities produced by **TRANSIMS** have been validated in the city of Portland; including (i) traffic invariants such as flow density patterns and jam wave propagation, (ii) macroscopic quantities, such as activities and population densities in the entire city, number of people occupying various locations in a time varying fashion, time varying traffic density split by trip purpose and various modal choices over highways and other major roads, turn counts, number of trips going between zones in a city, etc.

An Interaction Based Viewpoint. The **TRANSIMS** system has been designed using an interaction-based approach to capture the causes of observed traffic patterns. For each individual, his endogenous attributes are derived from the census data and his endogenous goals are derived from the activity patterns. His endogenous procedures or behavior consist of methods for finding specific locations to perform his desired activities, specific algorithms for finding routes to go from one location to another and specific rules used for driving. When such an endogenous individual interacts with the infrastructure and other individuals, we get traffic. The particular locations that an individual chooses, or the routes he takes are not determined solely by his endogenous attributes; they are a result of his goals, methods and his interaction with other individuals and the infrastructure. Similarly, the causal explanation of traffic or the question of who is at a given location at a given time, is given not only by the description of the individuals and the infrastructure, but also by the interaction amongst them. Thus consequences of large transformational changes such as a cascading power failure or infectious diseases can be understood in terms of the net effect of the interactions.

This is very different than traditional statistical models that fit parameters to given observations. Such systems that rely on observation and direct measurement of traffic cannot extrapolate into hypothetical scenarios precisely because they have no representation of the multitude of forces and interactions that lie behind each observation. As a simple example, the **TRANSIMS** methodology tells us how many people would be likely to use a new freeway if it were constructed. In doing so it captures what by now is well known as induced/latent demand. An observationally based system cannot extrapolate well beyond the circumstances in which it has been observed. Similarly, this approach will allow us to simulate the effects of changes in behavior or use of infrastructure on the overall social dynamics.

5.4 Telecommunication Sector

The telecommunication modeling environment is an extension of the **AdHopNet** [13, 24], designed to model extremely large, complex telecommunication networks made up of cellular networks, public switched telephone networks (PSTNs), Internet (IP) networks, and ad hoc mesh networks. It is an end-to-end simulation system, meaning that all aspects of the communication system are represented. Although simulations have been used for over four decades for representing and analyzing telecommunication systems, the use of high performance computing oriented simulations of very large telecommunication systems is a relatively new subject area; see [4, 29] for examples.

The system has been specifically designed to be interoperable with other infrastructure simulations and is useful for representing the complete system comprising the information and communication networks. It is also designed for technological scaling – as we move towards ubiquitous computing, telecommunication and computing networks with billions of heterogeneous transceivers. Such an integrated system can be used to evaluate federal policies on the use and operation of telecommunication infrastructures, especially in regards to potential effects of the policies on national security. It can also be used to discover and respond to new vulnerabilities that could occur while deploying adhoc and integrated networks, i.e., networks of mobile radio devices that present a constantly evolving telecommunication network.

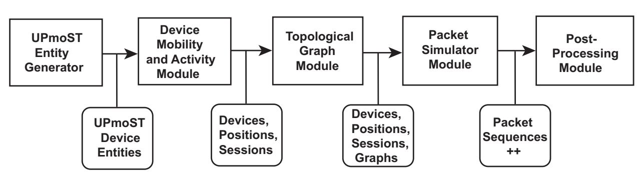

Fig. 12. Overall design of the telecommunication modeling module.

The modeling environment decomposes the telecommunication system into four basic time scales. The first module places devices and individuals throughout the urban region. It then generates the positions of transceivers at various times of the coarse simulation clock. This module also allows transceivers to become idle for some period of time and to rejoin the network at a later time. The module also provides for new transceivers to join the network and existing transceivers to leave the network permanently. Wireline devices are placed permanently at various locations based on the publicly available information.

In the second step, each device (e.g. phone, computers, etc) is assigned data sessions: the sessions are consistent with the kind of devices, their locations and their users. The sessions generated are statistically identical to the sessions generated in an urban region of interest. The next step consists of constructing a (time-varying) telecommunication network. Due to the various technologies used, these networks are dynamic and their topology varies significantly depending on the kind of technology used. This corresponds to intermediate time scale. Finally, at the finest time scale, voice or data is moved over the dynamic network; this aspect uses packet/voice data simulation methods based on flow techniques or discrete dynamical systems. The data is then stored succinctly using signal theoretic methods; Markov chain methods are then used to regenerate statistically equivalent packet streams. An auxiliary module is concerned with construction, analysis and regeneration of integrated telecommunication networks. The module synthesizes publicly available data sets in conjunction with population mobility information to construct the complete set of networks used in a telecommunication system: wireline, wireless, ad-hoc and the packet switched IP networks.

5.5 Public Health

The public health module (called **EpiSims**) of the integrated system simulates the spread of disease in urban areas. It details the demographic and geographic distributions of disease and provides decision makers with information about (1) the consequences of a biological attack or natural outbreak, (2) the resulting demand for health services, and (3) the feasibility and effectiveness of response options. See [22, 33, 34] for further details. **Simdemics**, an extension of **EpiSims**, is designed to model general reaction diffusion process such as vector borne diseases and simulation of social norms and fads.

Both **EpiSims** and **Simdemics** work by creating a social-network representing details of contacts between individuals based on their activity patterns which are provided by **TRANSIMS**. The system provides estimates of how disease will spread through a population depending on how it is introduced, how vulnerable people are, what responses are applied, and when responses are implemented.

The module simulates the movement of each individual from location to location in a large urban area as he or she goes about daily activities. The individuals are synthetic; they do not represent specific people, but a census taken on the entire synthetic population would be statistically indistinguishable from the actual census. On the other hand, the locations visited by individuals are real street addresses and reflect actual land-use patterns in the city.

The modeling environment associates a state of health with each individual being simulated. An individual's demographics determine his/her response to exposure and infection. For example, anyone over the age of 32 is assumed to have been vaccinated for smallpox. Exposure occurs in either of two ways: through contact with an infectious person or by visiting a contaminated location. The simulation user can introduce contamination at a location as an exogenous event in the simulation. Whether a person is infectious depends on when that person was exposed and their individual response to infection. By varying a few parameters, users can model many different diseases.

A simulated person's state of health may affect his or her actions. They may seek treatment at a nearby hospital or clinic, or they may stay home instead of pursuing certain activities. In addition, the user may specify actions that affect simulated people, such as mass or targeted vaccination/treatment/prophylaxis and isolation. Targeted responses are automated within the simulations: people are chosen at a user-specified rate from a list of symptomatic people; their contacts are found by following their schedule; and the contacts are then treated and/or isolated.

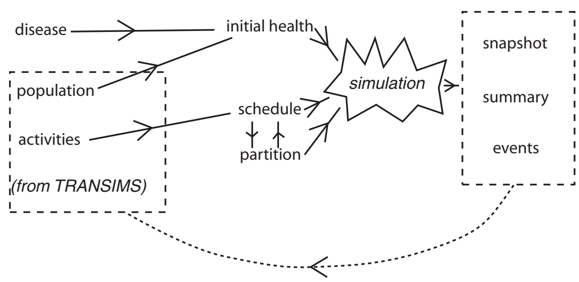

Fig. 13. Data flow in the epidemiology simulation system. Input data comes from two sources: the user's disease model and information about the social network. Stand-alone tools operate on the disease model and the population's demographics to produce the initial state of health for everyone in the simulation. Another tool converts a list of activities and locations organized by person into a schedule of events (primarily arrivals and departures) organized by location. The final preparation step estimates an optimal partition of resources among computational nodes. The simulation itself executes events in strict time order and propagates disease in accordance with the user's disease model.

5.6 Commodity Markets

Sigma is an agent-based, microscopic, computational modeling framework to study commodity markets. Systems such as **Sigma** offer several advantages to an economist interested in studying commodity markets, including (i) exact knowledge of what is exogenous and what is endogenous in the experiment, (ii) complete control on the amount of information accessible to the players, (iii) clear delineation of what information is public and private as well as what assumptions are reasonable to include.

The economist can not only study the system in equilibrium, but can also study the transient dynamics that lead to equilibrium conditions.

Sigma uses an interaction based computing approach to study the micro level behavior of the market and its players. The computational framework provides user, the ability to control individuals' preferences, behavior, market elements, trading mechanisms etc. This facilitates the study of different economic structures, strategies, policies and institutions in isolation. It can currently simulate a restructured electricity market. Three kinds of markets are modeled; centralized, decentralized, and a real-time (spot) market. The models employ economic theory-based methods and capture the dynamics of supply and demand in a market driven economy. New approaches that facilitate a wide range of experiments with a high degree of realism, include:

1. Flexible methods of aggregating individual consumers and producers into hierarchies in order to represent buyers and suppliers in residential, commercial, and wholesale markets
2. Heterogeneous demand profiles with elastic and inelastic components using time, location,activity, and demographic data for all individual consumers in a synthetic population
3. User-selectable economic clearing mechanisms to accommodate an array of market types, including Vickrey auction, double auction, and marginal price clearing.

The system simulates the activities (bidding, contracts, prices, etc.) of individual market players. The market model is driven by dynamic demand profiles that reflect the changing needs of individuals in an urban population. The model can be coupled to physical flow models for commodities that require physical clearing (such as electricity). The tool uses population dynamics and activity location data from a population dynamics simulation such as **TRANSIMS**. This information ties the market simulations to the urban infrastructure. Markets, among other things, are sensitive indicators of infrastructure disruptions and can be used to gauge public mood and awareness in crisis situations. The overall design of **Sigma** is depicted schematically in Figure 14. The framework, due to scaling requirements, has a parametric representation for buyers as well as sellers. This allows one to represent a number of realistic, individualistic, behavioral features that are typically assumed away in classical economic literature due to mathematical intractability. These include dropping classical Cournot oligopolists' assumptions, perfect rationality, information symmetry between consumers and generators, etc.

Sigma is a detailed simulation based analysis tool for simulating large commodity markets such as electricity markets. Markets are among other things, sensitive indicators of infrastructure disruptions and can be used to gauge public mood and awareness in crisis situations. The system can currently simulate large commodity markets such as the electric power market. It can be used to analyze effects of different regulatory changes, the impact of changes in consumer behavior on the clearing price, impact of price caps on demand and supply, market efficiency, generators' bid-

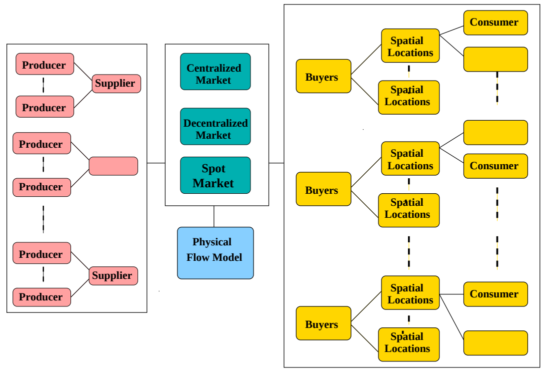

Fig. 14. Schematic diagram of the commodity market simulation system.

ding strategies etc. Another important use for such tools is their ability to analyze the effect of different market clearing rules on clearing prices.

The system simulates the activities (bidding, contracts, prices, etc.) of individual market players. The market model is driven by dynamic demand profiles that reflect the changing needs of individuals in an urban population. The model can be coupled to physical flow models for commodities that require physical clearing. The tool uses population dynamics and activity location data from a population dynamics simulation such as **TRANSIMS**. This information ties the market simulations to the urban infrastructure. The overall design of such a tool is depicted schematically in Figure 14. It consists of three main components that form a coupled system:

1. the electrical power grid, with associated elements including generators, substations, transmission grids and their related electrical characteristics.
2. a market consisting of market entities, including buyers, sellers, the power exchange (where electricity trades are carried out at various time/size scales), the independent system operator (ISO) and the market clearing rules and strategies.
3. an activity based individual power demand creator that yields spatio-temporal distribution of the power consumed.

Such simulations, due to scaling requirements, have a parametric representation for buyers as well as sellers. They allow for a number of realistic behavioral features that are typically assumed away in classical economic literature due to mathematical intractability. These include dropping classical Cournot oligopolist's assumptions, perfect rationality, symmetric information between consumers and generators, etc.

5.7 An Illustrative Use Case

The following use case built around **EpiSims** and **Simfrastructure** demonstrates how such modeling tools can be used for situational Awareness and consequence Analysis in the event of epidemics. In this scenario, during a heat wave in a city, terrorists shut down portions of the public transit system and a hospital emergency room during the morning rush hour. At the same time, they spread a harmless but noticeable aerosol at two commuter rail stations. These events, occurring nearly simultaneously, foster a chaotic, if not panic-stricken, mood in the general public.

EpiSims in conjunction with **Simfrastructure** can be used for situation assessment and consequence analysis. This is done by estimating the demand by demographics at emergency rooms and clinics under a variety of hypotheses to distinguish effects of the heat wave from those of a putative bio-attack. To accomplish this, several kinds of information is integrated: (i) population demographics and household structure, (ii) population mobility and transit timetables, (iii) hospital locations and capacities, (iv) natural history of various infectious diseases, (v) historical heat wave casualties, and (vi) (potential) surveillance data. We then estimate the demographics (age, gender, and home location) of people likely to have been in the two stations when they were "attacked". These are the people who would show up first for treatment if indeed a bio-attack had occurred. They also would serve as the subpopulation to seed with disease in a simulation. Biases in their demographics compared to a random sample of the population will induce persistent biases in the set of people infected at any time that cannot be captured by models assuming homogeneous mixing. We estimate demand at hospitals, assuming that people would arrive at a hospital near their home location. We further estimate whether each hospital had sufficient capacity to meet the demand. Historically, the most likely casualties of a heat wave are elderly people living alone with few activities outside the home. This information, combined with demographic and household structure data, allows us to estimate demand for health services created by the heat wave by demographic and location. For situation assessment, we note the obvious differences between these two demand patterns. In an actual event, comparison with admissions surveillance data would allow quick disambiguation between the two.

We estimate the likely spread of disease for several different pathogens by demographic and location. Furthermore, we can implement several suggested mitigating responses, such as closing schools and/or workplaces, or quarantining households with symptomatic people. Knowledge of the household structure permits an exceptionally realistic representation of the consequences of these actions. For example, if schools are closed, a care-giver will also need to stay home in many households. Or if households are quarantined when a member becomes symptomatic, we can estimate the immediate economic impact using the household incomes for exactly those households affected. Similarly, the economic impact of casualties with known demographics leads to a cost-benefit analysis for proposed interventions. In a similar study that we recently undertook, we found enormous differences in cost for interventions with similar numbers of casualties. Information on casualties can be fed back into the representation of the urban environment to evaluate effects on interdependent infrastructure.

The use case demonstrates the need for an interaction based modeling and simulation approach: such an approach captures physical inter-dependencies between infrastructures as well as implicit human-mediated interdependencies existing between infrastructures. For example, the demand for cooling on a hot summer day can strain the energy distribution system, forcing it to operate in a less robust regime.

Furthermore, the consequences of decisions made to mitigate accidents depend on the demand being serviced at the moment. Thus a decision to brown-out New York's financial district while maintaining service to residential areas has completely different effects at midnight on a Saturday than at 2 PM on a Wednesday. Practical decision support environments based on modeling environments such as **Simfrastructure** can evaluate such situation-dependent consequences.

6 Concluding Remarks

We described an interaction based approach to modeling and simulations of large scale socio-technical, biological and information systems. The theoretical foundations of this approach were based on sequential dynamical systems (SDS) and theory of large scale complex networks. Engineering principles are derived from such a theory. These engineering principles allow us to design simulations for extremely large systems and implement them on massively parallel architectures. As an illustration, we described **Simfrastructure**: a practical interaction based modeling tool to study large interdependent urban infrastructures. Large scale high performance computing oriented simulations for these systems are already operational; the simulations and the underlying systems would greatly benefit from further advances in interactive computing.

We are also currently exploring two broad research areas to further develop the interaction based design and analysis of extremely large heterogeneous systems: (i) discrete microscopic modeling and simulation of biological systems [52, 45, 54] and (ii) robust nanoscale design and computation.

Acknowledgements: We sincerely thank our colleagues, collaborators and the team members of the projects discussed here. Simfrastructure is being jointly developed with Karla Atkins, Keith Bisset, Richard Beckman, V. Anil Kumar, Achla Marathe, Henning Mortveit and Paula Stretz at Virginia Tech. The mathematical and computational theory of SDS was developed jointly with Harry B. Hunt III, S. Ravi, Daniel Rosenkrantz and Richard Stearns at University at Albany, Henning Mortveit at Virginia Tech and Christian Reidys at Los Alamos. The network theory is being jointly developed with Anil Vullikanti, Professor Aravind Srinivasan, Srinivasan Parthasarathy and Nan Wang at University of Maryland and Professors Ravi Sundaram (Northeastern) and Mayur Thakur (University of Missouri, Rolla).

References

1. Axtell R, Epstein J. Growing Artificial Societies: Social Science From the Bottom Up MIT Press / Brookings Institution
2. Albert R, Barabasi A (2002) Statistical mechanics of complex networks, Rev. Mod. Phys. 74, pp. 47-97

3. Atkins K, Barrett C, Homan C, Marathe A, Marathe M, Thite S (2004) Marketecture: A Simulation-Based Framework for Studying Experimental Deregulated Power Markets, 6th IAEE European Conference, Modeling in Energy Economics and Policy, Zurich
4. Bajaj L, Takai M, Ahuja R, Bagrodia R (1999) Simulation of Large Scale Communication Systems. Proc. MILCOM'99
5. Barabási A, Albert R (1999), Emergence of scaling in random networks. Science, 286
6. Barrett C, Eubank S, Marathe M, Mortveit H, Reidys C (2002) Science and Engineering of Large Scale Socio-Technical Simulations, Proc. 1st International Conference on Grand Challenges in Simulations held as a part of Western Simulation Conference, San Antonio Texas, 2002, (2002).
7. Barrett C, Eubank S, Anil Kumar V, Marathe M (2004) Understanding Large Scale Social and Infrastructure Networks: A Simulation Based Approach, in SIAM news 37(4), Appears as part of Math Awareness Month on The Mathematics of Networks.
8. Batty M (2005) Hierarchy in Cities and City Systems, CASA Working paper series 85-11-4. To appear in Hierarchy in natural and Social Sciences, D. Pumain, Ed., Kluwer Academic Publishers
9. Barrett C, Beckman R, Berkbigler K, Bisset K, Bush B, Campbell K, Eubank S, Henson K, Hurford J, Kubicek D, Marathe M, Romero P, Smith J, Smith L, Speckman P, Stretz P, Thayer G, Eeckhout E, Williams M (2000) TRANSIMS: Transportation Analysis Simulation System. Technical Report LA-UR-00-1725, Los Alamos National Laboratory Unclassified Report, 2001. An earlier version appears as a 7 part technical report series LA-UR-99-1658 and LA-UR-99-2574 to LA-UR-99-2580.
10. Barrett C, Reidys C (1999) Elements of a Theory of simulation I: Sequential CA over random graphs Appl. Math. and Comput., 98:241–259
11. Barrett C, Mortveit H, Reidys C, (2000) Elements of a Theory of simulation II: Sequential dynamical systems. Appl. Math. and Comput., 107(2-3), pp. 121–136
12. Istrate G, Hansson A, Thulasidasan S, Marathe M, Barrett C, (2005) RESTORED: A Methodology for Semantic Compression of TCP Traces. Technical Report, Los Alamos National Laboratory, submitted
13. C. Barrett, M. Marathe, H. Mortveit, S. Ravi, C. Reidys, J. Smith. AdHopNet: Advances in Simulation-based Design and Analysis of Ad-Hoc Networks Technical Report LA-UR 00-1567, Los Alamos National Laboratory, 2000.
14. Barrett C, Marathe M, Smith J, Ravi S, (2002) A mobility and traffic generation framework for modeling and simulating ad hoc communication networks. ACM Symposium on Applied Computing (SAC), pp. 122-126
15. Barrett C, Hunt III H, Marathe M, Ravi S, Rosenkrantz D, Stearns R, (2001) Analysis Problems for Sequential Dynamical Systems and Communicating State Machines. Proc. International Symposium on Mathematical Foundations of Computer Science (MFCS'01), Czech Republic, pp. 159–172
16. Beckman R, Baggerly K, McKay M (1996) Creating synthetic base-line populations, Transportation Research Part A – Policy and Practice 30, pp. 415–429
17. Buss S, Papadimitriou C, Tsitsiklis J (1991) On the Predictability of Coupled Automata: An Allegory About Chaos. Complex Systems, 1(5), pp. 525–539
18. Brand D, Zafiropulo P (1983) On Communicating Finite State Machines. J. ACM, 30(2), pp. 323–342
19. Barrett C, Hunt III H, Marathe M, Ravi S, Rosenkrantz D, Stearns R, Tosic P, (2001) Gardens of Eden and fixed points in sequential dynamical systems. Proc. of the International Conference on Discrete Models - Combinatorics, Computation and Geometry (DM-CCG), Paris, pp. 95–110

20. Barrett C, Hunt III H, Marathe M, Ravi S, Rosenkrantz D, Stearns R, Thakur M, (2004) Complexity of Predecessor Existence Problems for Finite Discrete Dynamical Systems. Technical report, Virginia Tech. Preliminary Version appeared in Discrete Models of Complex Systems (DMCS) 2002.
21. Barrett C, Hunt III H, Marathe M, Ravi S, Rosenkrantz D, Stearns R (2003) Reachability Problems for Sequential Dynamical Systems with Threshold Functions. Theoretical Computer Science, 1-3, pp. 41-64
22. Barrett C, Smith J, Eubank S, (2005) Modern Epidemiology Modeling, in *Scientific American*
23. Broder A, Kumar R, Maghoul F, Raghavan P, Rajagopalan S, Stata R, Tomkins A, Wiener J, (2000) Graph structure in the web. Computer Networks, 33
24. Barrett C, Marathe M, Mortveit H, Reidys C, (2001) SDS Based Specification of Large IP Networks Technical Report, Los Alamos National Laboratory
25. Barrett C, Drozda M, Engelhart D, Anil Kumar V, Marathe M, Morin M, Ravi S, Smith J (2005) Structural Analysis of Ad Hoc Networks: Implications for Protocol Performance, to appear in IEEE International Conference on Wireless and Mobile Computing, Communications and Networking
26. Carley K, Fridsma D, Casman E, Altman N, Chang J, Kaminski J, Nave D, Yahja A, (2003) BioWar: Scalable Multi-Agent Social and Epidemiological Simulation of Bioterrorism Events. NAACSOS conference proceedings, Pittsburgh, PA
27. Breiger R, Carley K, Eds., (2003). NRC workshop on Social Network Modeling and Analysis, pp. 133-145, National Research Council
28. Chung F, Lu L (2002) Connected components in a random graph with given degree sequences. Annals of Combinatorics, 6:125–145
29. Cowie J, Nicol D, Ogielski A (1999) Modeling 100 000 Nodes and Beyond: Self-Validating Design. DARPA/NIST Workshop on Validation of Large Scale Network Simulation Models
30. Creignou N, Khanna S, Sudan M, (2001) Complexity Classifications of Boolean Constraint Satisfaction Problems SIAM Monographs on Discrete Mathematics and Applications
31. DYNAMIT, 1999. DYNAMIT. Massachusetts Institute of Technology, Cambridge, Massachusetts. `http://mit.edu/its/dynamit.html`
32. DeHon A (2002) Very Large Scale Spatial Computing Proc. of the Third International Conference on Unconventional Models of Computation (UMC'02), 15-19, pp. 27–37
33. Eubank S, Guclu H, Anil Kumar V, Marathe M, Srinivasan A, Toroczkai Z, Wang N (2004) Modeling Disease Outbreaks in Realistic Urban Social Networks, Nature, 429, pp. 180-184
34. Eubank S, Anil Kumar V, Marathe M, Srinivasan A, Wang N (2004) Structural and Algorithmic Aspects of Large Social Networks, Proc. 15th ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 711-720
35. Eubank S, Anil Kumar V, Marathe M, Srinivasan A, Wang N, (2005) Structure of Social Contact Networks and their Impact on Epidemics. to appear in AMS-DIMACS Special Volume on Epidemiology
36. Floreen P, Orponen P (1994) Complexity Issues in Discrete Hopfield Networks Comp. and Learning Complexity of Neural Networks: Advanced Topics, Edited by I. Parberry
37. Epstein J (2005) Generative Social Science: Studies in Agent-Based Computational Modeling, Princeton University Press, Forthcoming
38. Foster I and Kesselman G, Eds. (2000) The Grid: Blueprint for a New Computing Infrastructure.

39. Faloutsos C, Faloutsos P, Faloutsos M (1999) On Power-Law Relationships of the Internet Topology, Proc. ACM SIGCOMM
40. Garzon M, (1995) Models of Massive Parallelism: Models of Cellular Automata and Neural Networks, EATCS Monographs on Theoretical Computer Science, Springer Verlag
41. Goldin D, Smolka S, Attie P, Sonderegger E (2004) Turing Machines, Transition Systems, and Interaction Information and Computation, 194(2), pp. 101-128
42. Gutowitz H, Ed. (1989) Cellular Automata: Theory and Experiment North Holland
43. Harel D, Kupferman O, Vardi M (1997) On the complexity of verifying concurrent transition systems. Proc. 8th Int. Conf. Concurrency Theory (CONCUR'97), Warsaw, Poland, July 1997, Vol. 1243 LNCS, pp. 258–272
44. Haimes Y, Horowitz B (2004) Modeling Interdependent Infrastructures for Sustainable Counterterrorism. Journal of Infrastructure Systems, pp. 33-41
45. Hartwell L, Hopfield J, Leibler S, Murray A, (1999) From Molecular to Modular Cell Biology, Nature, Vol. 402, pp. C47-C51
46. Hethcote H (2000) The Mathematics of Infectious Diseases SIAM Review, 42(4):599–653
47. Hillel B (1996) Cities as Movement Economies. Urban Design International 1, pp. 49-60
48. Hopfield J (1982) Neural Networks and Physical Systems with Emergent Collective Computational Abilities. Proc. of National Academy of Sciences of the USA, No. 81, pp. 3088-3092
49. Hunt III H, Stearns R, Marathe M (2000) Relational Representability, Local Reductions and the Complexity of Generalized Satisfiability Problem. Technical Report No. LA-UR-00-6108, Los Alamos National Laboratory.
50. Hunt III H, Stearns R, Marathe M (2001) Strongly Local Reductions and the Complexity/Efficient Approximability of Algebra and Optimization on Abstract Algebraic Structures, International Conference on Symbolic and Algebraic Computations (ISAAC), pp. 183-191
51. Hunt III H, Rosenkrantz D, Barrett C, Marathe M, Ravi S (2001) Complexity of Analysis and Verification Problems for Communicating Automata and Discrete Dynamical Systems. Technical Report No. LA-UR-01-1687, Los Alamos National Laboratory
52. Ideker T, Galitski T, Hood L (2001) A New Approach to Decoding Life, Ann. Review of Genomics and Human Genetics, Vol. 2, pp. 343-372
53. Istrate G, Marathe M, Ravi S (2001) Adversarial models in evolutionary game dynamics. Proc. 12th ACM-SIAM Symposium on Discrete Algorithms (SODA'2001), Washington, DC, pp. 719–720
54. Kitano H (2002) Computational Systems Biology, Nature, 420, pp. 206-210
55. Kleinberg J (2000) Navigation in a Small World. Nature 406
56. Kung H (1982) Why Systolic Architectures. IEEE Computers, 15(1), pp. 37-42
57. Little R (2002) Controlling Cascading Failure: Understanding the Vulnerabilities of Interconnected Infrastructures. Journal of Urban Technology, 9(1), 109-123
58. Liu R, Lloyd E, Marathe M, Ramanathan R, Ravi S (2005) Algorithmic Aspects of Topology Control Problems For Ad-hoc Networks, in ACM/Baltzer J. Mobile Networks and Applications (MONET), 10, pp. 19-34
59. Liu M, Subramaniam M (2005) An Approach for Formal Specification and Verification of Large Scale Simulation Systems Using Abstract State Machines, Los Alamos National Laboratory Technical Report, 2005.
60. Laubenbacher R, Pareigis B (2003) Decomposition and simulation of sequential dynamical systems, Advances in Applied Math, 30, pp. 655-678

61. Lee E, Wallace A, Mitchell J, Mendon D, Chow J, (2003) Managing disruptions to critical interdependent infrastructures in the context of the 2001 WorldTrade Center attack Beyond September 11: An account of post-disaster research M. F. Myers, Ed. Boulder, CO: Natural Hazards Research and Applications Information Center, University of Colorado, Program on Environment and Behavior, Special Publication 39, pages 165-198
62. Margolus N (2000) An Embedded DRAM Architecture for Large-Scale Spatial-Lattice Computations Proc. 27th Annual International Symposium on Computer Architecture (ISCA), pp. 149-158
63. Margolus N, Toffoli T Cellular Automata Machines: A New Environment for Modeling, Cambridge: MIT press, 1987.
64. Milner R (1999) Communicating and Mobile systems: the π-calculus Cambridge University Press
65. Milner R (1993) Elements of Interaction, C.ACM, 36:1, pp. 78-89
66. Moore C (1990) Unpredictability and Undecidability in Dynamical Systems. Physical Review Letters, 64(20), pp 2354-2357
67. Mortveit H, Reidys C (2001) Discrete, Sequential Dynamical Systems. Discrete Mathematics, 226:281–295
68. Nagel K, Schrekenberg M A Cellular Automata Model for Freeway Traffic. J. de Physique I, France, 2:2221, 1995.
69. Nagel K, Stretz P, Pieck M, Leckey S, Donnelly R, and Barrett C (1998) TRANSIMS Traffic Flow Characteristics. Transportation Research Board Annual Meeting.
70. Nagel K, Paczuski M (1995) Emergent traffic jams. Phys. Rev. E, 51:2909-2918
71. Newman M, (2003) The structure and function of complex networks SIAM Review, 45, 167–256
72. Nicthiu C, Remila E (1998) Simulations of graph automaton. Proc. Mathematical Foundations of Computer Science (MFCS'98) Satellite workshop on graph automata, Th Worsch and R. Wolmar Eds, Universität Karlsruhe, pp. 69-78
73. Orponen P (1994) Computational Complexity of Neural Networks: A Survey, Nordic Journal of Computing, 1(1), pp. 94-110
74. Rabinovich A (1997) Complexity of Equivalence Problems for Concurrent Systems of Finite Agents. Information and Computation, 127(2), pp. 164–185
75. Raney B, Cetin N, Völlmy A, Vrtic A, Axhausen K, Nagel K (2003) An Agent-Based Microsimulation Model of Swiss Travel: First Results, Networks and Spatial Economics, 3, pp. 23–41
76. Schreckenberg M, Schadschneider A, Nagel K, Ito N (1995) Discrete stochastic models for traffic flow. Physical Review E, 51:29-39
77. Speckman P, Vaughn K, Pas E (1997) Generating Household Activity-Travel Patterns (HATPs) for Synthetic Populations. Transportation Research Board 1997 Annual Meeting.
78. Speckman P, Vaughn K, Pas E (1997) A Continuous Spatial Interaction Model: Application to Home-Work Travel in Portland, Oregon. Transportation Research Board 1997 Annual Meeting
79. Sheffi Y (1985) Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming methods. Prentice Hall
80. Stearns R, Hunt III H (1996) An algebraic model for combinatorial problems. SIAM J. Computing, 25, No. 2, 448-476
81. Shukla S, Hunt III H, Rosenkrantz D, and Stearns R (1996) On the Complexity of Relational Problems for Finite State Processes. International Colloquium on Automata Programming and Languages (ICALP), pp. 466-477

82. Tripp J, Hansson A, Gokhale M, Mortveit H (2005) Partitioning Hardware and Software for Reconfigurable Supercomputing Applications: A Case Study, Proc. Supercomputing SC'05.
83. Watts D, Strogatz S (1998) Collective dynamics of 'small-world' networks, Nature, vol. 393, pp. 440-442
84. Wegner R (1998) Interactive Foundations of Computing, Theoretical Computer Science, 192, pp. 315-351
85. Wolfram S, Ed. (1987) Theory and applications of cellular automata, World Scientific
86. http://dimacs.rutgers.edu/Workshops/Opening/abstracts.html
http://www.mel.nist.gov/div826/msid/sima/simconf/march04/phsi.pdf
http://www.iupui.edu/ ilight/symposium04/ILIGHTPPT/measrespon.pdf
http://www.mgmt.purdue.edu/centers/seas/Research/Research.htm.