A science for measuring cities: hybrid indicator frameworks for the smart city

Author List Hidden

ABSTRACT

For a given project or policy in a city, what is the "right" key performance indicator framework to evaluate its impact to the city? At the very least, the right indicator framework should be determined by the trajectories of the indicators within that framework. But there are a variety of constraints that we can impose on those trajectories; the contribution of this paper is to convert knowledge about the indicators into constraints on the trajectories. We develop a mathematical theory for mapping certain classes of complex systems models, from simple 'co-design' models to more complicated cyber-physical systems models, to systems of key performance indicators, and call such models *hybrid indicator frameworks*. Finally, we illustrate the framework by applying it to two smart-city projects in Portland, Oregon, and Hampton Roads, Virginia.

1. INTRODUCTION

City administrators choose *key performance indicators*, or KPIs, as a part of a larger effort to understand, communicate, and realize their strategic priorities. There are several strategy-setting frameworks for choosing KPIs, from balanced scorecards [13] to SMART [12] to more specialized urban planning frameworks; in such frameworks, the KPIs are often designed in tandem with new projects, new policies, and new processes. Once chosen, they are used in program evaluation and reporting, in departmental reviews, and in request-for-proposal creation.

More recently, the advent of "smart cities" and the Internet of Things have given city administrations an opportunity to reform and to clarify their strategic priorities. As public services from energy grids to transportation networks to healthcare systems are automated, cities are becoming cyber-physical systems (CPS) in the true sense of the term—systems whose physical processes are profoundly integrated with computation and data. But a smart city is more than just the sum of many smart services; it the sum of many smart services and all the ways in which they interact with each other, both physically and computationally. By understanding these interactions—and the data that they generate—we can make better decisions about the services themselves, e.g. about how to optimize existing services or how to solicit new solutions. In order to understand these interactions and the data that they generate, our indicator frameworks for evaluating data must incorporate CPS models of the interactions.

A generic indicator is just a consistent source of data. Instead of beginning with the indicators, this paper begins conceptually with *indicator frameworks*: assemblages of simple indicators, however defined, which measure both physical and computational processes. The meaning of "measure" will be made precise below; measurement is defined in terms of the "trajectory" (or, more simply, "up" versus "down") for any indicator. The processes themselves may be modeled at various levels of abstraction. In this paper, we use a relatively simple co-design model based on work by [10], leaving other, more general CPS models to future work.

Our work is driven by practical considerations in project governance at the Global City Teams Challenge [26], a U.S. federal government smart city program led by National Institute of Standards and Technology (NIST) with participation from over 120 cities and 300 organizations worldwide. For example, consider a city that is deploying two projects: one which improves transit signal priority so that travel times for people are reduced and another which increases and improves crosswalks with smart lighting to reduce traffic-related fatalities. If people feel safer, there could be a positive net gain in individuals who will also ride transit and travel times could be reduced further showing a co-benefit or positive trajectory on both indicators. However, depending on the actual design of the roadway corridor, increased pedestrian crossings could impede traffic as transit always has to stop and travel times may stay neutral or even increase, showing an unexpected negative effect on a separate indicator. Co-benefits across indicators can be useful in helping invest in the right projects. Conversely, if two projects have opposite effects on two indicators, then a city can understand why they did not see the effect the planned project was projected to have. If such interactions can be identified beforehand, then an improved set of indicators can be developed and/or the project design can be adjusted.

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

2. BACKGROUND

Modern cities are composed of several highly networked cyber-physical systems from the electric Grid to the water supply to transportation Networks. These are integrated together to provide services essential for city residents. Individually, each service or city process can be thought of as a coupled feedback system where a software subsystem controls a physical subsystem by collecting data on it through sensors and providing the necessary control action. For example, the heating system of a building measures the temperature of the building via sensors and then controls the heater state. The output from the software subsystem is assumed to pass through the actuation subsystem, which converts the software output into control actions (mechanical inputs) and then imposes them onto the physical subsystem. The control action changes with the state of physical subsystem. Consequently, the state of the physical subsystem changes with time due to the control action from the software subsystem. Thus, each CPS in essence is a hybrid system with both discrete and continuous dynamics.

Compared to other cyber-physical systems such as robotics, manufacturing, and even aviation, cities are more complex, more diverse, and more unpredictable. This is due to the fact that cities are a composition of these multi-domain CPS, wherein the interaction between each of the domain is not strictly engineered as in the subcomponents of a robotic system or an aircraft. In a traditional engineering domain like robotics, the CPS can be analyzed using dynamical systems theory, with the assumption that the systems are typically in equilibrium and they can be controlled with the typical tools of planning and management. However, in a city system, which is influenced by social physics and where the interactions are orchestrated via humans, the traditional models are often found lacking [9, 8]. This is primarily due to the unpredictable nature of humans, whose behavior ranges from irrational and unpredictable at best, to intelligent, deceptive, and malicious at worst [22, 20]. Cities are notoriously hard to model and comprehend with closed form analytical methods. The sheer scale, range of actions and diversity adds another level of complexity.

For example, the Architecture Analysis and Design Language (AADL) [15, 14] is a standard developed by the Society of Automotive Engineers. Originally developed for aerospace systems, the standard is applicable to the model-based specification and analysis of embedded real-time systems and systems of systems. It has comprehensive support for modeling a variety of component types and their interactions. Component abstractions in AADL consist of software components, computational hardware, and the overall system. Different interaction patterns between components are supported in AADL. Using AADL it is possible to conduct analysis for a variety of critical system properties, such as performance, schedulability and reliability. Other general-purpose approaches similar to AADL are OMG's SysML and OMG's MARTE profile for UML. SysML [17] is a general-purpose modeling language for systems engineering. SysML leverages a subset of the Unified Modeling Language (UML) while extending it with capabilities needed to model complex systems engineering problems, which is called the SysML profile for UML. The extensions enable engineering analysis. The Modeling and Analysis of Real-time and Embedded (MARTE) systems [23] is a UML profile to extend UML to support the model-driven development of real-time and embedded systems. However, it is difficult to use these modeling and description languages for cities because of uncertainty in subsystem interactions.

Without such integrated models it is difficult to understand the integrated performance impact of various decisions taken at the level of different city systems. Therefore, traditionally city based systems have been studied using simulations, game-theoretic models and agent-based models. Game-theoretic models [25, 27] describe the problem by modeling the individual agents and describing their interactions as negotiations, and then using several mechanisms to arrive at a globally optimal solution. Initially, game theoretic methods assumed that all agents worked together collaboratively, but they have also been formulated to use an adversarial model, especially in economics, business, and market strategy. Another approach to studying and modeling these types of problems is agent-based modeling and simulation [19, 9], which uses individual, communicating agents to evaluate system properties. This is the approach used in the MATSim ("Multi-Agent Transport Simulation Toolkit") toolkit [7, 18]. This toolkit implements a multi-agent transport simulator that can be used to describe the activities of agents, which in turn provides a demand model. Combined with a set of origination/destination maps and capacity plans, the toolkit can simulate the system performance.

Another alternative is to use a data-driven, human-in-the-loop approach to analyzing cities. For example, [32] describes how the data collected from various public transportation vehicles can be used to analyze the performance of transit systems. The authors describe a data-driven performance measurement tool for waste management in [34]. [33] describes various performance indicators for low carbon cities in china. The data-driven approach is also useful for better understanding system anomalies and behaviors [24, 11]. System experts and engineers can use the information gleaned from this data to update operations procedures and even redesign some of the city systems. Data-driven approaches have now been adopted almost universally across major cities in the world, and have influenced the creation of many indicator frameworks, from CITYKeys [1] in the European Union to ISO/TS 37151 [2] by the International Standards Organization. Such indicator frameworks, however, are essentially ad hoc: they give a light-weight classification of indicators along hierarchies, and focus much more on justifying the inclusion of individual indicators into a framework than their large-scale interrelation.

We subscribe to a data-driven approach insofar as we begin with a technical definition of indicator framework. Concretely, however, it is the *lack* of sufficient data that has driven our inclusion of CPS and other tools from dynamical systems theory into indicator frameworks. The processes of a city are simply too complex to measure directly. If we had the data, we would just generate a correlation matrix with it. Unfortunately, we do not, so we must develop models to make up for that deficit.

3. HYBRID INDICATOR FRAMEWORKS

Indicators convey information in a consistent way across time and across different systems. They may be sensors, scientific measurements, or arbitrary database columns, so long as they can "indicate"—i.e., go up or down. Indicators may be also associated to processes; to say that an indicator is a "key performance" indicator is just to state that the indicator is associated to some given process or processes. For the purposes of this paper, we will represent indicators as objects in a given sort of *category*, and trajectories as finite sequences valued in partially-ordered sets.

3.1 Preliminaries

First, some definitions.

A *category* $\mathcal{C}$ is defined by the following data:

1. A collection of objects, called the objects of $\mathcal{C}$.
2. A collection of maps between any two objects $c, c' \in \mathrm{Ob}\,\mathcal{C}$, called the arrows from c to c', e.g. $f : c \to c'$.
3. A composition rule between arrows, written $g \circ f$ for first f then g, satisfying an identity axiom and an associativity axiom.

A *functor* from a category $\mathcal{C}$ to a category $\mathcal{D}$ is a mapping $F : \mathcal{C} \to \mathcal{D}$ that associates each object in $\mathcal{C}$ to an object in $\mathcal{D}$, and associates each arrow $f : c \to c'$ to an arrow $F(f) : F(c) \to F(c')$ in such a way that respects the composition rule.

Category theory is a branch of mathematics originally used to translate results from one area of mathematics to another, e.g. from topology to algebra. It has now been applied to subjects ranging from quantum mechanics [3, 6] to dynamical systems modeling [30, 16] to database theory [28] to general models of processes [5] and, most recently, to cyber-physical systems [31]. For an exposition of category theory, we recommend the reader to any of the introductory texts for computer scientists [4], engineers [29], or for mathematicians [21].

A *partially-ordered set*, or poset, is a set with a relation $\leq$ satisfying reflexivity, anti-symmetry, and transitivity. A monotone map between posets is an order-preserving function, i.e. f is monotone iff $a \leq b \Rightarrow f(a) \leq f(b)$. An antitone function is an order-reversing function, i.e. $a \leq b \Rightarrow f(a) \geq f(b)$.

A *(co)-design problem* is a certain kind of optimization problem involving *functionalities* F and *resources* R, with the constraint that F and R are posets. Each design problem, dp, represents an optimization problem for a particular process whose inputs are resources in R and whose outputs are functionalities in F. Importantly, primitive design problems can be composed to create larger (co-)design problems by series, parallel, and loop operations.

The expression $[\mathrm{dp}](\mathsf{f}, \mathsf{r})$ may be thought of as a statement of the form, "given functionality f and r, is there an implementation that delivers f at cost r?" We refer to [**?**] for the details. In this paper, we will mainly use the fact that design problems form a category, DP, whose objects are posets and arrows are profunctors

$$[\mathrm{dp}] : \mathsf{F}^{\mathrm{op}} \times \mathsf{R} \to \mathsf{Bool}.$$

In particular, we will be analyzing co-design diagrams in the form of Figure 3.1.

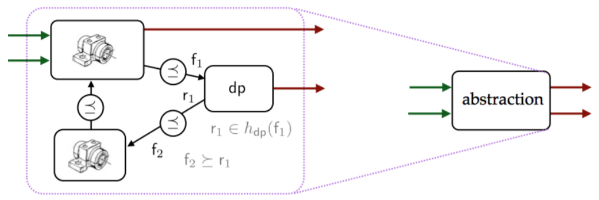

Figure 1: Multiple design problems compose to form one co-design problem.

3.2 Indicator Frameworks

Definition 1 *An* indicator framework *is a small category $\mathcal{C}$ with the following conventions:*

1. *An object c in $\mathcal{C}$ is a Boolean set $\{\uparrow, \downarrow\}$ called an* indicator *in $\mathcal{C}$.*
2. *Each arrow of $\mathcal{C}$ is a function $f : c \to c'$ called a* (simple) correlation *from c to c'.*
3. *The identity map $\mathrm{id}_c : c \to c$ is called the* primary id *on c. On most indicators, the primary id is time.*
4. *An object c in $\mathcal{C}$ with no outgoing arrows is called an* indicator type.

A trajectory *on $\mathcal{C}$ is a functor $T : \mathcal{C} \to \mathsf{Poset}_W$, where Poset_W is the category of posets and monotone maps between posets satisfying a "windowing" condition: indicator values can be mapped to other indicator values only if their time values are within some ϵ-window of each other.*

For any indicator $c \in \mathcal{C}$, an element $x \in T(c)$ is called an indicator value.

There is an analogy between the above and the definition of a database schema in [28]; this is to be expected since most indicator frameworks are, essentially, views into a database. The difference is that we are not tracking parent-child relations in the sense of the foreign keys of a database, but linear relations in the sense of indicators going 'up' and 'down'.

Indicator frameworks can be graphed as categories, while trajectories can be written down as collections of tables. As shorthand, we use green arrows to represent positive correlations $\uparrow \mapsto \uparrow$, and red arrows to represent negative correlations $\uparrow \mapsto \downarrow$, as in the following example:

$$\begin{tikzcd}
\text{Traffic Volume} \arrow[d] \arrow[dr] & \text{Wind Speed} \arrow[d] \\
\text{Mobile Sources} \arrow[r] & \text{NO}_2 \text{ Levels}
\end{tikzcd}$$

Traffic Volume	
Time	**Indicator Value**
09:00:00 10-13-2016	High
10:00:00 10-13-2016	Medium
11:00:00 10-13-2016	Low

NO_2 Levels	
Time	**Indicator Value**
09:00:00 10-13-2016	11 ppb
10:00:00 10-13-2016	10 ppb
11:00:00 10-13-2016	7 ppb

Each correlation is, in effect, a constraint on the possible trajectories of the indicators. To see this, recall that a trajectory must satisfy the formal definition of a functor. In the example, this amounts to showing that the diagram below (and others like it) commutes: $T \circ f = T(f) \circ T$.

$$\begin{tikzcd}
\text{Traffic Volume} \arrow[r, "f"] \arrow[d, mapsto, "T"'] & \text{NO}_2 \text{ Levels} \arrow[d, mapsto, "T"] \\
(H, M, L) \arrow[r, "T(f)"'] & (11, 10, 7)
\end{tikzcd}$$

Fix some $\epsilon > 0$. In the case above, the only monotone map satisfying the windowing condition is $(H \mapsto 11, M \mapsto 10, L \mapsto 7)$. It exists, so the constraint is satisfied. Similarly, if f had been a negative correlation, we would have had to find an antitone map satisfying the windowing condition. Note that as we increase the ϵ-window, we always increase the number of possible monotone and antitone maps. This is important in case our data values do not match very nicely in time, but if we take it too far, our ϵ-windowed correlations become useless.

Vice versa, given a set of times and indicator values—i.e. a mapping $T : \mathrm{Ob}\,\mathcal{C} \to \mathsf{Poset}_W$—we can also check whether those indicator values satisfy the constraints implied by the presence of the arrows in the indicator framework.

Definition 2 *A* hybrid indicator framework *is a functor from any category to an indicator framework $\mathcal{C}$.*

The key point here is that *any* model which can be formulated as a category can be used to power a hybrid indicator framework. In the rest of this section, we will build an example using the category of design problems, DP, as our modeling category.

Recall our motivating question: what is the right set of indicators to evaluate *a given solution*? We have chosen design problems as our modeling category in particular because a design naturally posits the possible trajectories of a given complex system—a natural dynamical system concept—as possible *solutions* to that system. Hybrid indicator frameworks may be thought of as answering the opposite question: given a possible solution as input, or rather a whole space of possible solutions in the form of a co-design diagram, what are the possible trajectories of our indicators, given how we expect them to be correlated?

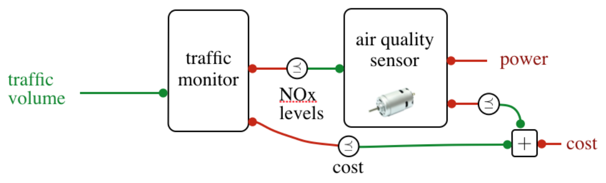

Figure 2: A simple co-design diagram for a sensor and traffic monitor.

As mentioned before, design problems form a category, DP, whose objects are posets (interpreted as resources) and arrows are design problems. Suppose that one of the functionalities provided by a traffic sensor is a pedestrian count, which is then consumed by a traffic computer. The sensor and the traffic computer are processes, and the indicator value of that sensor is a functionality provided by the sensor, and a resource consumed by the traffic computer. Formally, a co-design diagram thinks of each process—what goes inside the box—as a profunctor to Bool (essentially, this is a Bool-valued square matrix with columns all the possible values of the sensor), but we will not really use this fact.

In other words, the productive use of constructing a co-design diagram is that it forces us to consider the interaction between processes.

Intuitively, the relationship between a co-design diagram $\mathcal{D}$ and an indicator framework $\mathcal{C}$ is simple: each wire in the diagram corresponds to an indicator in $\mathcal{C}$, and each box attached to the wire corresponds to some process associated with that indicator. More formally, for any given trajectory T on $\mathcal{C}$, a functor H from DP to $\mathcal{C}$ is defined by

$$\begin{aligned} P \in \mathrm{Ob}\,\mathsf{DP} \quad &\longmapsto \quad c \in \mathrm{Ob}\,\mathcal{C} \text{ such that } T(c) = P \\ \mathrm{dp} : P \to P' \quad &\longmapsto \quad f : P \to P' \end{aligned}$$

One can check that this definition respects the composition in dp. If we interpret correlations in $\mathcal{C}$ as constraints on the possible trajectories of indicator values, then literally the mapping above is lifting design problems—which are used to model interactions (in this case, simple input-output interactions) between processes—to constraints on the trajectories.

4. APPLICATIONS

Hybrid indicator frameworks are general enough to support many existing indicator frameworks for cities, including examples such as CITYKeys [1] or ISO/TS 37151 [2] where the key contextual structure is simply hierarchical grouping; one only needs to write down a category that captures the hierarchical structure, e.g. an operad.

The remainder of this section is dedicated to exploring two ongoing implementations of the framework: Portland, Oregon and Hampton Roads, Virginia.

4.1 Portland, Oregon

The City of Portland (specifically, the Bureau of Planning and Sustainability and the Bureau of Transportation) is investigating how networks of low-cost air quality sensors can be used to improve real-time understanding of transportation-related pollution along roadway corridors. Using such data, Portland would like to evaluate the impacts on air quality and climate-related variables from transportation solutions such as improved transit signal priority, improved traffic signal coordination, increased electrification of the urban fleet, and truck priority signaling.

Reductions in traffic-related pollution and greenhouse gas emissions are not the only indicators Portland needs to evaluate. Portland's transportation policies are also aiming to reduce traffic-related fatalities, travel delay for people and freight, and understand how changes in the transportation system affect gross regional product. These indicators of interest are components of multiple systems/models present in the urban environment including urban air quality, transportation network, human activity patterns, freight network and the economy. A final outcome/next step of this paper's work will be to determine (a) how the Portland performance indicators are impacted directly by the various transportation projects (positively, negatively, or no effect) and (b) how the indicators are impacted by effects that are a result of a change in another indicator, thus defining how the indicators are related to each other.

In order to move towards this overlay of multiple models and identify the interactions between indicators across several urban systems, first a diagram of the urban air quality system focusing on roadside nitrogen oxides (NOx) concentrations was compiled based on knowledge from the subject matter expert, an air quality scientist, shown in Figure 3. This diagram serves several purposes; identify all of the possible variables or processes that can change the concentrations of roadside NOx, identify which variables can or cannot

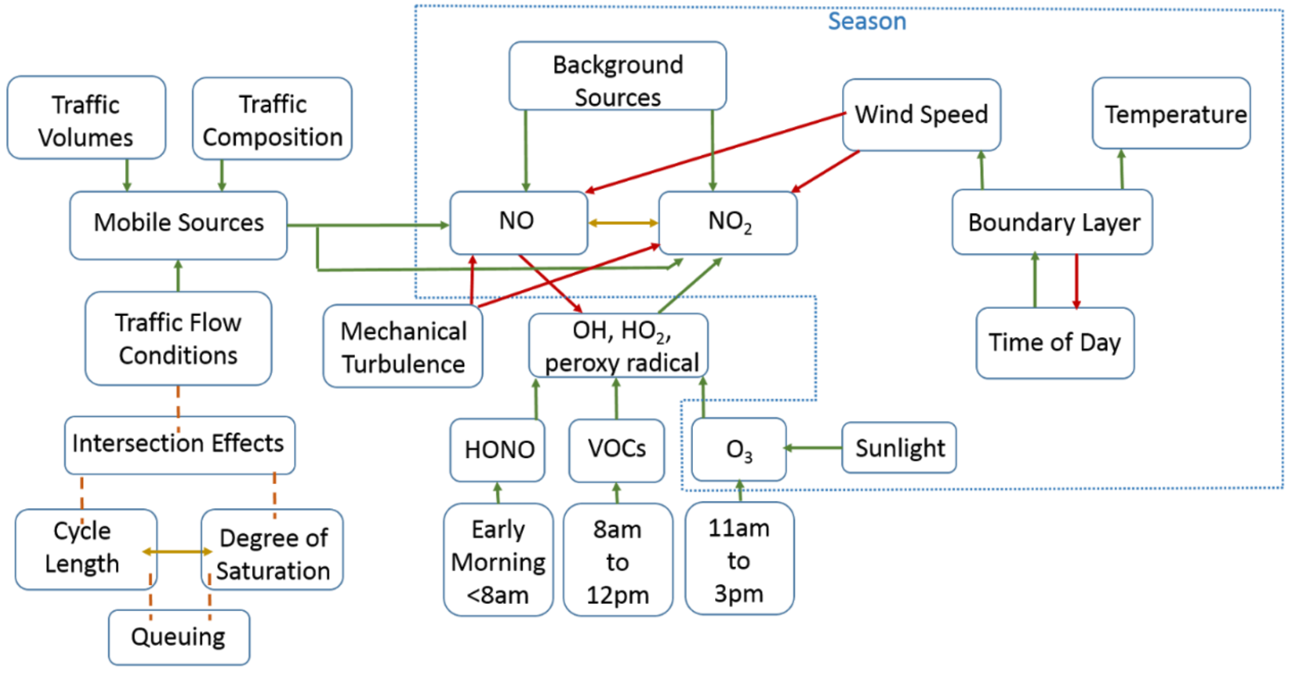

Figure 3: Roadside NOx conceptual model with hypothesized relationships to variables that may alter roadside NOx concentrations. Green represents a positive correlation, red represents a negative correlation, orange represents inconclusive.

be measured, and which variables have existing data sources that can be leveraged.

In Figure 3, NO, NO_2, O_3, wind speed, sunlight, and temperature are all variables that can be assessed with roadside sensor and background instrument measurements. Other variables like HONO and VOCs are not species that are measured at the roadside or even at background monitoring stations regularly but time of day could be used as a proxy to understand certain trends in NO and NO_2. Background monitoring stations can be used to assess background concentrations of NO and NO_2 which will not provide an estimate of emissions as a result of specific background sources but can still provide some information on regional levels of pollution separate from the localized roadway's mobile source contribution.

So far, we have only described an indicator framework, not a hybrid indicator framework. The next step is to integrate the diagram above with a co-design diagram. This consists in defining relationships between (a) Portland's transportation solutions and indicators and (b) between the indicators themselves, i.e. to produce a similar diagram as Figure 3 but focused in on the traffic volume, composition, and traffic signal system side and other components of the transportation system compiled by a transportation subject matter expert.

4.2 Hampton Roads, Virginia

The following example is derived from the StormSense group, composed of the Virginia Institute of Marine Science (VIMS) at the College of William and Mary and many of the cities in the Greater Hampton Roads Region of Tidewater Virginia, including: Newport News, Virginia Beach, Norfolk, Hampton, Chesapeake, Portsmouth, Williamsburg, and York County.

Hampton Roads, Virginia, is deploying a network of IoT-enabled water-level sensors, called StormSense, in order to enhance its emergency preparedness for extreme and recurrent flooding. Developed by VIMS, StormSense is a complex system that integrates these indicators with urban-scale hydrodynamic flood modeling and forecasting in the Chesapeake Bay. These models are then used to enhance emergency preparedness for flooding resulting from storm surge, rain, and tides.

The StormSense instance of the hybrid indicator framework centers around a generic smart city functioning as a networked cyber-physical system informed by flood predictions from a hydrodynamic model up to 36 hours prior to an inundation event. In this framework, depicted in Figure 4, all components and their constituent variables (represented alongside arrows with units; as appropriate) associate with each other and are locally and globally networked in the co-design diagram. The essential elements of the StormSense flood modeling hybrid indicator framework relate to many divisions of the modern-day smart city, their constituent citizen stakeholders, natural systems and the academic research at the cross-section of these systems (color coded regions of Figure 4).

5. DISCUSSION

In this paper, we have developed a simple, categorical approach for combining indicator frameworks and complex system models. We have done so with the goal of predicting and measuring the impacts of a smart-city solution to a city, including possible unintended consequences. The framework

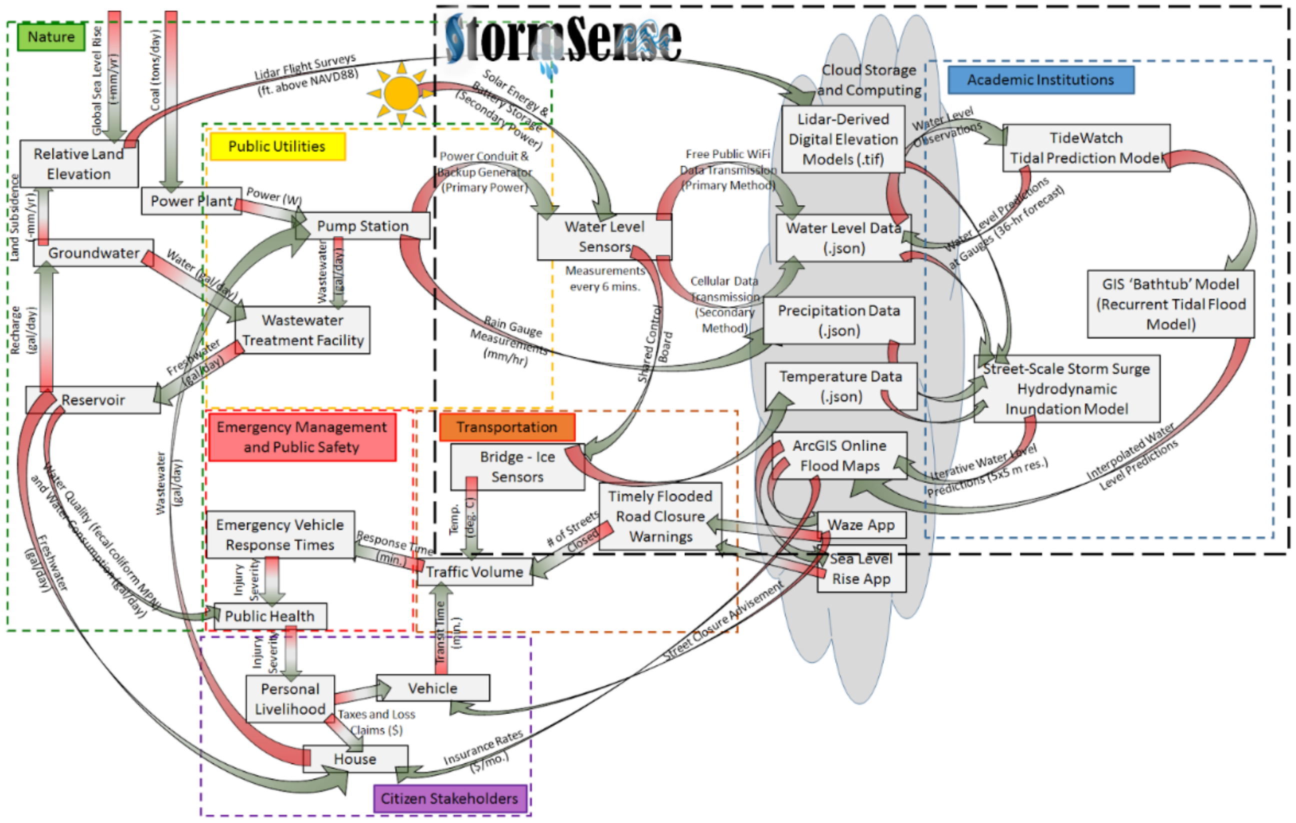
StormSense
Nature
Public Utilities
Emergency Management and Public Safety
Transportation
Citizen Stakeholders
Academic Institutions
Cloud Storage and Computing
Relative Land Elevation
Power Plant
Groundwater
Reservoir
Pump Station
Wastewater Treatment Facility
Water Level Sensors
Lidar-Derived Digital Elevation Models (.tif)
Water Level Data (.json)
Precipitation Data (.json)
Temperature Data (.json)
ArcGIS Online Flood Maps
Waze App
Sea Level Rise App
TideWatch Tidal Prediction Model
GIS 'Bathtub' Model (Recurrent Tidal Flood Model)
Street-Scale Storm Surge Hydrodynamic Inundation Model
Bridge - Ice Sensors
Timely Flooded Road Closure Warnings
Emergency Vehicle Response Times
Traffic Volume
Public Health
Personal Livelihood
Vehicle
House
Global Sea Level Rise (+mm/yr)
Coal (tons/day)
Lidar Flight Surveys (ft. above NAVD88)
Solar Energy & Battery Storage (Secondary Power)
Power (W)
Power Conduit & Backup Generator (Primary Power)
Free Public WiFi Data Transmission (Primary Method)
Cellular Data Transmission (Secondary Method)
Measurements every 6 mins.
Rain Gauge Measurements (mm/hr)
Shared Control Board
Water Level Observations
Water Level Predictions at Gauges (36-hr forecast)
Iterative Water Level Predictions (5x5 m res.)
Interpolated Water Level Predictions
Land Subsidence (-mm/yr)
Recharge (gal/day)
Water (gal/day)
Wastewater (gal/day)
Freshwater (gal/day)
Water Quality (fecal coliform MPN) and Water Consumption (gal/day)
Freshwater (gal/day)
Response Time (min.)
# of Streets Closed
Temp. (deg. C)
Injury Severity
Transit Time (min.)
Taxes and Loss Claims ($)
Insurance Rates ($/mo.)
Street Closure Advisement

Figure 4: Theoretical co-design diagram depicting a flood prediction methodology with hypothetically-related variables in a generalized smart city. Color coded regions represent interrelated elements of inundation prediction and disaster management. Red sections of arrows represent outputs, while green arrows represent inputs; direction of arrows suggest general flow gradient of information or resources.

was designed to provide technology providers and end-users, including local governments, a way to identify the relevant key performance indicators and to assess in advance the changes to the indicators after the deployment of solution. By using complex system models of city processes and their interactions, we can identify relevant indicators even as systems scale up in complexity. Two examples from real-world smart-city projects were introduced to demonstrate the applicability of the method, and they showed how the process of identifying relationships between indicators can be decomposed through the creation of a diagram and then the mapping of that diagram to an indicator framework.

At this point, the relationships of indicators are described only at the conceptual level, and correlations were defined with only a simplified "up-down" semantics. Obviously, the relationships between indicators could be nonlinear, and additional analysis and research will be necessary to suggest a more concrete method for formulating such relationships. In the case that the relationships are nonlinear or explicit solutions cannot be found, simulation-based methods may be used to track and assess the movements of the KPIs. We expect that the system modeling and decomposition approach used here will need to be further refined and reformulated.

Concretely, we expect to implement and verify hybrid indicator frameworks over the next year in a subset of the 100+ projects in NIST's Global City Teams Challenge. The proposed indicator framework will be used to assess the impacts of the solutions to the city and communities, and the validity of the framework will be verified by comparing its predictions against the actual data collects after each projects' completion.

6. ACKNOWLEDGEMENTS

We would like to thank David Spivak and Andrea Censi for conversations on category theory, as well as Bilin and Levent Guvenc for their contributions to the examples.

7. REFERENCES

[1] Citykeys.
[2] I. 37151:2015. Smart community infrastructures – principles and requirements for performance metrics. Technical report, 2015.
[3] S. Abramsky and B. Coecke. Categorical quantum mechanics. *ArXiv e-prints*, Aug. 2008.
[4] S. Awodey. *Category Theory*. Oxford Logic Guides, 2006.
[5] J. C. Baez and B. Fong. A Compositional Framework for Passive Linear Networks. *ArXiv e-prints*, Apr. 2015.

[6] J. C. Baez and M. Stay. Physics, Topology, Logic and Computation: A Rosetta Stone. *ArXiv e-prints*, Mar. 2009.

[7] M. Balmer, K. Meister, M. Rieser, K. Nagel, K. W. Axhausen, K. W. Axhausen, and K. W. Axhausen. *Agent-based simulation of travel demand: Structure and computational performance of MATSim-T*. ETH, Eidgenössische Technische Hochschule Zürich, IVT Institut für Verkehrsplanung und Transportsysteme, 2008.

[8] M. Batty. Building a science of cities. *Cities*, 29:S9–S16, 2012.

[9] M. Batty, K. W. Axhausen, F. Giannotti, A. Pozdnoukhov, A. Bazzani, M. Wachowicz, G. Ouzounis, and Y. Portugali. Smart cities of the future. *The European Physical Journal Special Topics*, 214(1):481–518, 2012.

[10] A. Censi. A Mathematical Theory of Co-Design. *ArXiv e-prints*, Dec. 2015.

[11] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[12] G. T. Doran. *Management review*, 70(11):35–36, 1981.

[13] M. J. Epstein and J.-F. Manzoni. The balanced scorecard and tableau de bord: translating strategy into action. *Strategic Finance*, 79(2):28, 1997.

[14] P. Feiler, B. A. Lewis, and S. Vestal. The SAE Architecture Analysis & Design Language (AADL) A Standard for Engineering Performance Critical Systems. In *Computer Aided Control System Design*, pages 1206–1211, 2006.

[15] P. H. Feiler, D. P. Gluch, and J. J. Hudak. The Architecture Analysis & Design Language (AADL): An Introduction. Technical Report ADA455842, DTIC Document, 2006.

[16] B. Fong. The Algebra of Open and Interconnected Systems. *ArXiv e-prints*, Sept. 2016.

[17] M. Hause et al. The SysML Modelling Language. In *Fifteenth European Systems Engineering Conference*, volume 9, 2006.

[18] J. Illenberger, G. Flotterod, and K. Nagel. Enhancing matsim with capabilities of within-day re-planning. In *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pages 94–99, Sept 2007.

[19] S. Karnouskos and T. N. De Holanda. Simulation of a smart grid city with software agents. In *Computer Modeling and Simulation, 2009. EMS'09. Third UKSim European Symposium on*, pages 424–429. IEEE, 2009.

[20] P. Liu and Z. Peng. China's smart city pilots: A progress report. *Computer*, 47(10):72–81, 2014.

[21] S. Mac Lane. *Categories for the Working Mathematician*. Springer Science+Business Media, 1971.

[22] S. Munir, J. A. Stankovic, C.-J. M. Liang, and S. Lin. Cyber physical system challenges for human-in-the-loop control. In *Presented as part of the 8th International Workshop on Feedback Computing, Berkeley, CA*, 2013.

[23] Object Management Group. *UML Profile for MARTE: Modeling And Analysis of Real-Time Embedded Systems, Version 1.1*. Object Management Group, OMG Document formal/2011-06-02 edition, June 2011.

[24] S. J. Qin. Survey on data-driven industrial process monitoring and diagnosis. *Annual Reviews in Control*, 36(2):220–234, 2012.

[25] L. J. Ratliff, M. Jin, I. C. Konstantakopoulos, C. Spanos, and S. S. Sastry. Social game for building energy efficiency: Incentive design. In *Communication, Control, and Computing (Allerton), 2014 52nd Annual Allerton Conference on*, pages 1011–1018, Sept 2014.

[26] S. Rhee. Catalyzing the internet of things and smart cities: Global city teams challenge. In *2016 1st International Workshop on Science of Smart City Operations and Platforms Engineering (SCOPE) in partnership with Global City Teams Challenge (GCTC) (SCOPE - GCTC)*, pages 1–4, April 2016.

[27] J. S. Rosenschein and G. Zlotkin. *Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers*. MIT Press, Cambridge, MA, USA, 1994.

[28] D. I. Spivak. Functorial Data Migration. *ArXiv e-prints*, Sept. 2010.

[29] D. I. Spivak. *Category Theory for Scientists*. MIT Press, 2014.

[30] D. I. Spivak and J. Z. Tan. Nesting of dynamic systems and mode-dependent networks. *ArXiv e-prints*, Feb. 2015.

[31] D. I. Spivak, C. Vasilakopoulou, and P. Schultz. Dynamical Systems and Sheaves. *ArXiv e-prints*, Sept. 2016.

[32] F. Sun, Y. Pan, J. White, and A. Dubey. Real-time and predictive analytics for smart public transportation decision support system. In *2016 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 1–8, 2016.

[33] C. Williams. Measuring in all the right places: Themes in international municipal eco-city index systems. 2014.

[34] A. U. Zaman and S. Lehmann. *Journal of Cleaner Production*, 50:123–132, 2013.