# Modelling interfaces in the decentralized ecosystems of the Internet of Things

*Completed Research Paper*

**David Pym**
University College London
Gower Street, London, UK
d.pym@ucl.ac.uk

**Will Venters**
London School of Economics
Houghton Street, London, UK
w.venters@lse.ac.uk

## Abstract

*The Internet of Things concerns extensive ecosystems of 'things' connected by interfaces. Researching such ecosystems is challenging given that devices extend into multiple administrative domains with different boundaries and communications structures. In this paper, we explore how formalism, based on systems thinking and mathematical logic, might be used within a critical realist study of a sociotechnical IoT ecosystem. Specifically, we draw upon theories of distributed systems and their models to define IoT things in terms of locations, resources, and processes. Next, we explain how a formal logic can be used define and reason about IoT things and interfaces, explaining also the logically defined concepts of compositionality and local reasoning. Finally, we show how such models can be used to study a specific IoT ecosystem in order to derive generative mechanisms relevant to the socio-technical complexity of IoT ecosystems. In this way, we explore how we might introduce formal modelling within a critical realist epistemology.*

**Keywords:** Systems Modelling, Interfaces, Logic, Process Algebra, Critical Realism, Internet of Things, IoT Systems, Compositionality, Local Reasoning.

## Introduction

The Internet of Things (IoT) concerns the 'pervasive presence around us of a variety of *things* or *objects*… which are able to interact with each other and cooperate with their neighbors to reach common goals' (Atzori et al. 2010, p. 2786, italic in original). It has been defined as 'a world-wide network of interconnected objects uniquely addressable, based on standard communications protocols' (Gubbi et al. 2013) . Such objects, or things, might include sensors, actuators, mobile phones, RFID tags and the context of their use includes cities (e.g., Smart City initiatives), industry, and the home (e.g., Home Automation) and are expected to have profound effects, particularly when they are coupled with cloud computing enabled data storage and processing (Botta et al. 2016). Such things reside on local networks that communicate with other local networks and devices, either directly ('peer-to-peer') or via network-based servers, through agreed standardised interfaces. The IoT has come 'to represent complex systems and systems of systems' that may 'transform capitalist economies' yet it poses 'complex challenges' including that 'devices and services tend to coexist within systems of systems that might have porous boundaries' with severe security and safety limitations in the technical realization of such systems of systems (Nicolescu et al. 2018). From the perspective of computer science and the theory of systems, we can understand the Internet of Things within an established framework (Anderson and Pym 2016; Caulfield and Pym 2015; Collinson et al. 2012; Collinson and Pym 2009) for conceptualizing, modelling, and reasoning about *distributed systems* (Coulouris et al. 2011). Indeed, in computer science, the concept of a distributed system is the fundamental building block of the systems theory that underpins the technical architecture and operations of the Internet and the services — including, for example, the distributed databases and banking systems that support online commerce — that it supports. The core concepts of distributed systems are about the following:

- architecture: where are the components of the system,
- resource: what are the components — that is, the things — of the system, and
- process: how do the components of the system communicate with one another, and how does the system deliver its intended services.

Central then to IoT is the concept of an *object or thing*[1] and its *interface* with other things. Yet such digital technology things and interfaces are poorly theorized within our field of information systems (Orlikowski and Iacono 2001) . Here IoT *things* introduce boundaries that may be political, interpretive and contested as they form part of complex distributed ecosystems (and platforms) that often extend into multiple administrative domains (Barrett et al. 2015; Henfridsson and Bygstad 2013; Tiwana et al. 2010) through various forms of boundary resources including interfaces and standards (Ghazawneh and Henfridsson 2013; Karhu et al. 2018). Our ability to provide situated accounts of IoT in practice should thus capture the complex meshing of technical interactions, dependencies, interoperabilities, and institutional relations (Kallinikos 2004) which constitute their distributed ecosystems. For example, an IoT household water meter might be associated with a householder and their water supplier but installed in a separately owned space (e.g., the pavement), managed by a different firm, communicate via a mobile standard (e.g., NB-IoT) and associated billing system, and be contested in the validity of its readings (e.g., it might leak). Its operation might be dependent upon other devices, cloud services and systems, and upon other companies (such as those who change its batteries).

Our most popular Information Systems epistemological tools are, however, somewhat limited in their ability to provide situated accounts of such interfaced and enmeshed IoT. Dominant empiricist approaches, such as positivism, seek regularities and patterns of generalization that deny the extended systems of relations and connections that make each IoT use in practice unique. Similarly, interpretivist and social constructivist approaches may prove misleading by not sufficiently accounting for the agency of the technology involved in distributed ecosystems particularly of the complex digital interactions inherent within their technical interfaces. One proposed response has been to examine services (Mathiassen and Sørensen 2008) or platforms (Constantinides et al. 2018) which abstract such distributed systems in innovative ways, or to seek to incorporate materiality within ontologically constructivist accounts (e.g., sociomateriality).

Critical realism (CR) has been posited to overcome the shortcomings of these extremes by acknowledging a realist ontology to 're-establish a realist view of being in the ontological domain whilst accepting the relativism of knowledge as socially and historically conditioned in the epistemological domain' (Mingers 2004 p91). By seeing mechanisms that cause events which may be observed, while accepting that social phenomena are socially constructed, CR provides explanation through causal mechanisms that enable generalizations beyond a specific case. We note, however, that causality is contingent, 'placing the same technology in a different context does not imply that the same mechanisms will be activated' (Klecun et al. 2014, p. 151). Deriving such causal mechanisms through CR requires some form of understanding IoT's distributed ecosystems of connections and interfaces. It is to this task that our paper is addressed. Systems thinking provides an approach to such understanding (Mingers 2011), albeit, ironically, systems thinking has been somewhat absent within much information *systems* research (Alter 2004; Demetis and Lee 2016). This is surprising when systems thinking was central to our disciplines creation as it emerged from cyberneticians' attempts to model complex systems (for a history see: Checkland 1981; Pickering 2010).

This paper contributes by developing a formal systems model of distributed IoT systems based on mathematical modelling techniques (Anderson and Pym 2016; Caulfield and Pym 2015; Collinson et al. 2012) and focused on the technical elements of IoT things and interfaces. Crucially, within this modelling we introduce *compositionality* which allows models to be logically separated with defined interfaces between them. Along with compositionality comes a principle of *local reasoning*, which supports analyses of systems component-by-component. These two features allow the modelling of IoT systems of systems and thus of boundaries and decomposition.

[1] We consider here only things within an Internet of Things definition (though our logical modelling could be used for more generic modelling of things in general).

Next, the paper outlines how such models might be practically used, as part of a critical realist perspective, to provide a situated account of IoT ecosystems. Specifically, how formal modelling of compositionality might be used to model IoT ecosystems within CR studies in order to derive generative mechanisms. This may be of both practical and epistemological value as formal models can be rendered in software and may be formally verified using automatic analysis tools (O'Hearn 2019). This leads us to discuss how CR might reveal the situated practice of IoT by isolating discrepancy against such software-based logical models – discrepancy that may be researched through more traditional CR epistemological approaches.

We summarize now the organization of the remainder of this paper. In the next section, we explain why we believe mathematical modelling of IoT systems can be helpful, referring to Soft Systems Methodology, and discussing the methodology by which models are constructed. Then, in the following section, we describe a compositional mathematical modelling theory for distributed systems, based on concepts of location, resource, and process, together with an associated system of logic that supports local reason about the components of models. In particular, we emphasize the key concepts of compositionality, interfaces, and especially local reasoning. Next, we give an extensive explanation, with developed examples, of the role and use of the modelling of IoT ecosystems within the context of research in information systems. We conclude with a discussion of the significance of modelling for information systems research in the IoT, and discuss some directions for further work.

## Why introduce mathematical modelling of IoT systems?

As IoT 'things' exist within complex extended systems which often remain hidden we argue that it would be helpful to have a framework for deriving mathematical models of technical things and their interactions against which empirical findings based might be compared within a CR study. Such modelling is particularly useful for IoT things which may be physical (e.g., embedded in a road) or virtual (a software object processing data flows) yet in achieving their 'common goals' they exist within complex extended systems of communication and coordination that are hard to observe but which have considerable real-world consequences.

Modelling systems within subjectivist ontologies is evident within methodologies such as 'Soft System Methodology' (SSM)(Checkland 1981). In Figure 1, we compare SSM's interpretive modelling approach (Checkland and Holwell 1998) with a standard modelling cycle from within computer science. In both observations of the perceived real world (1) lead to the derivation of models (2) which are used to compare to empirical observations in order to derive interpretations (3) and thus used to act upon the world with the hope of improving it in the next iteration (4).

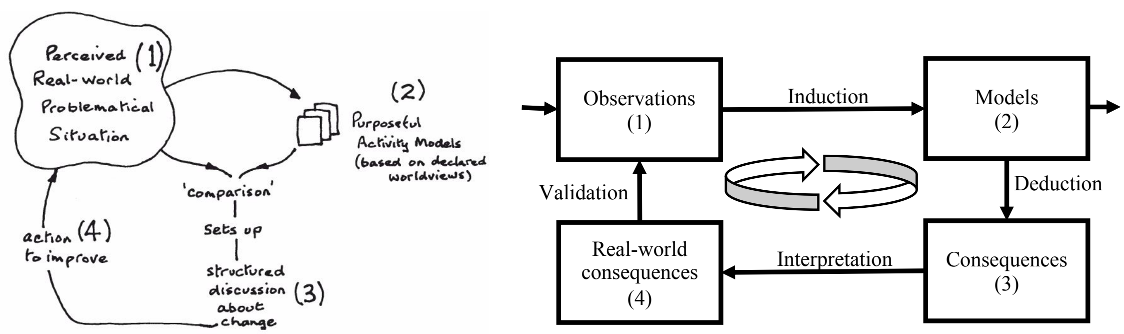


**Figure 1. SSM's modelling cycle (left, from Checkland and Holwell (1998)) and a standard modelling cycle (right).**

Within our critical realist approach to IoT studies, and in contrast to SSMs soft-systems modelling as simply epistemic devices, we believe our formal models should be reductionist and so seek to understand minimal structures within the ecosystem that capture the technical properties of interest in any given context. The process described in Figure 1 leads to the construction of mathematical objects that capture the empirical observations of the 'real-world'. The mathematical objects that are constructed have the same status as any other abstract mathematical construct and can be analyzed in the usual mathematical ways. Nevertheless, their structures reflect the real world, though are inhibited by our limitations in observing the real world. As such, models can inform and bound the empirical observations derived from our worldly reality.

However, we also acknowledge the systems perspective that 'the whole is more than the sum of its parts' (Checkland 1981) and so such models must be considered in relation to the wider socio-technical IoT ecosystem. In order to understand what this might coherently mean in the world of mathematical modelling, we need to focus on what is meant by 'the sum of parts': in our setting the summing operation is between component submodels and requires a notion of interface between those sub-models in order to describe their interaction.

This leads us to develop *compositionality* as a key feature of our models: that is, that a model of a relatively complex systems consists in a *composition* of models of simpler systems (reflective of the compositionality evident in ecosystems of IoT systems). The idea derives from the 'Principle of Compositionality' or 'Frege's Principal', after the German logician, mathematician, and philosopher Gottlob Frege. This principle asserts that the meaning of the whole system should be determined by the meanings of its parts. In modelling terms, this amounts to a requirement that the functionality of a composite model should be determined by the functionalities of its component models. Moreover, when reasoning about models we seek to reason about the component models *independently* of one another and to reason about the overall model by combining, with due care and attention, the reasoning about the components. We call this principle *local reasoning*. Local reasoning is vital to support conceptual and computational tractability in modelling.

If a model is a composition of component models, how, then is the composition of two formal models to be defined? The situation is quite delicate. First, systems do not exist in isolation; rather, systems are parts of ecosystems of systems with which they interact. Typically, however, we neither can nor want to model the entire ecosystem. We may wish to model just a single system or a small fragment of the ecosystems. In either case, we must introduce into our modelling a framework a concept of *environment*; that is, that part of the ecosystem that is not part of the system of interest. Individual models must therefore be able to interact with other models and with their environment. In either case, these interactions must be specified using a concept of *interface*. Interface thus becomes a significant component of compositionality and of the ability to understand IoT ecosystems from a systems perspective.

In Simon's *The Sciences of the Artificial* (Simon 1996), we find the following:

> *An artifact can be thought of as a meeting point — an 'interface' in today's terms — between ''inner' environment, the substance and organization of the artifact itself, and an 'outer' environment, the surroundings in which it operates.*

Here, we consider a class of artefacts (in Simon's sense) that, while in fact quite generic, are inspired by the IoT concept and by distributed systems within computer science (e.g., Coulouris et al. 2011) as consisting of a number of independent computers (the Things) linked by a network (the Internet). More technical computer science definitions (see, for example, the Wikipedia entry for 'Distributed computing') go further:

> *'A distributed system is a system[2] [that is, assembly of components] whose components [things] are located on different networked computers [that is, the physical devices], which communicate and coordinate their actions by passing messages to one another. The components [things] interact with one another in order to achieve a common goal.'*

More simply, we wish to build a formal minimal model of IoT technologies 'mathematical reality' within a stratified critical realist ontology such that comparison may be made in the domain of the empirical (our observed events/interpretations) to better identify events and mechanisms by discrepancies.

Our goal in this paper, then, is to define from a systems perspective, in formal and logical detail, the concept of *thing* and of *interface* such that we might compare IoT *in use* with such formal models in order to hypothesize mechanisms reflective of the use of the IoT in practice and to reason about — ideally using local reasoning (O'Hearn et al. 2001) — its key properties, in areas such as performance, security, and privacy.

[2] A generic definition will suffice at this point.

## A framework for modelling things and interfaces

Our systems modelling approach — as described in (Collinson et al. 2012), building on theoretical work as described in (Anderson and Pym 2016; Collinson and Pym 2009), and deployed in (Caulfield and Pym 2015) — builds on the conceptualization of distributed systems that we have described above; that is, we consider the architecture of a system, the components (or things) in the system, and the processes that are executed by the system. A simple abstraction identifies the concepts *location*, *resource*, and *process* — as well as *environment* — as being the key components of systems models. Given these, we can derive, that is construct, a concept of *interface*.

For our account of how to build mathematical models of distributed systems and, in particular, the Internet of Things, this latter definition is more helpful because it draws out some key concepts.

- First, the idea that the *things* — that is, the system's resources — are *located*.
- Second, that there are, in general, multiple *networks* of resources.
- Third, that these located components perform *actions* in order to *communicate* with one another.

While our focus here is on IoT, distributed systems are more pervasive including the Internet itself, intranets, online gaming, global banking transactions, and cloud computing infrastructure (where a key problem is the maintenance of consistency with different copies while simultaneously maintaining the robustness of the service).[3]

From the perspective of mathematical modelling, a very simple abstraction gives a very direct way to begin to model distributed systems and thus the IoT. We thus base our model of IoT on the following key concepts:

- *Locations*: a collection of linked places, be they physical or virtual, that constitutes the basic architecture of a system. Individual IoT devices, computers, file stores, and so on exist at locations within a distributed system. (Note locations might also be places within computers at which the CPU (Central Processing Unit) and other components reside);
- *Resources*: the entities that a system uses — consumes, creates, moves — in the course of its operations. Examples of resources include the memory locations at which data is stored, the processor cycles available to perform computations, and human operators required to manage and maintain systems; and
- *Processes*: the collection of activities, which are mostly concurrent, that constitutes a system's operations, and so delivers its services. Examples of services include a website, streaming films, reporting water-usage, and the multitude of system-level services provided by a computer's operating system in order to perform computations, manage inputs and outputs, manage a memory and storage, send and receive messages, etc..

Additionally, a specific system, described using these components, resides within an *environment*, and resource transfers between the system and its environment characterize the service that the system provides.

Mathematically, we may use the following structures to model our set-up:

- For *locations* and their interconnections, we might work with *directed graphs*; that is, collections of vertices (the locations) connected by edges (the connections) that are directed or oriented (and may 'go both ways'). For example, consider Figure 2, in which we have a directed graph that has two separate components and in which some edges go both ways and some go just one way.

[3] For example, the CAP (Consistency, Availability, Partition-tolerance) theorem in the theory of distributed systems, also known as Brewer's theorem (https://en.wikipedia.org/wiki/CAP_theorem), establishes that only two of the CAP properties can be maintained simultaneously. Handling this problem is important in maintaining the experience of many cloud computing services.

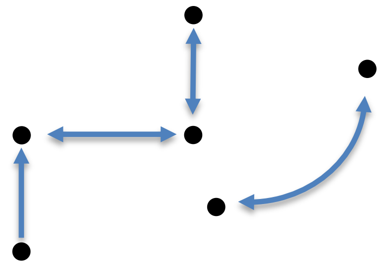

**Figure 2. Directed graphs**

- For *resources*, we can define *an algebra of resource elements*. A simple example of an algebra of resources is one that permits resource elements to be *combined* and *compared*. Consider for example, an algebra for natural numbers with 0, addition, and less-than-or-equals, written:

  $$(\mathbb{N}\,,0\,,+\,,\leq)$$

  Such an algebra describes resources that are counted using the natural numbers ℕ, such as money, or the seats in the rows of a theatre. For example, 3 dollars + 5 dollars is less than 10 dollars, and one can do more with 10 dollars than with 3 dollars + 5 dollars; and 4 dollars + 6 dollars is equal to 10 dollars, and one can do the same with 10 dollars as with 4 dollars + 6 dollars. More abstractly, we employ structures called ordered partial monoids (O'Hearn and Pym 1999; Pym 2019), which generalize this view to account for richer situations than just arithmetic. For example, another, and very important, example of such a structure is given by the memory cells of computer RAM (Ishtiaq and O'hearn 2001), leading to Separation Logic (Ishtiaq and O'hearn 2001; Reynolds 2002) which is the basis of program analysis tools such as Facebook's 'Infer'.[4]
- For *processes* — that is, actions and communications — we propose using a *synchronous algebra of concurrent processes*, such as something resembling Milner's SCCS (Milner 1983). Complex processes, that describe the execution paths of systems and the communications between systems. Algebraically, complex processes, written *E*, *F*, etc., are built up from *basic actions* using *combinators* that describe operations such as the following:

  - *sequencing* of actions, written as $a : b : c : \ldots$ , in which $a$ is followed by $b$ is followed by $c$, and so on; for example for an IoT water-meter we may define,

    $$\mathit{receiveWaterLevel} : \mathit{CompareWithPreviousLevel} : \mathit{ReportWaterUsage}$$

    and, in general, a basic action can be followed by fully general process, including all the combinators we describe below, and written as $a : E$
  - *choice* between processes $E$ and $F$, written as the process $E + F$, so that either the process $E$ or the process $F$ executes; for example,

    $$\mathit{ResetWaterLevel} : \mathit{Continue} + \mathit{SaveWaterLevel} : \mathit{Continue}$$

    where *Continue* is the name of the ongoing process that manages meters
  - *concurrent composition* of processes $E$ and $F$, written as the process $E \times F$, so that the processes $E$ and $F$ execute together;[5] for example,

    $$\mathit{ReportWater} : \mathit{Continue} \times \mathit{ResetCounter} : \mathit{Continue}$$

[4] https://engineering.fb.com/developer-tools/open-sourcing-facebook-infer-identify-bugs-before-you-ship/

[5] Concurrent execution may be either synchronous or asynchronous. This topic is a vast one in the mathematical theory of computation, but, for now, we assume that we are working with synchronous concurrent composition, in which the corresponding actions in each process execute at the same time. See Milner (1983) for a clear exposition of the issues.

- and a range of combinators that variously handle things like hidden or local actions, recursion, and processes that denote no action and the termination of a process.

These components provide a structured account of what Simon's definition describes as the 'inner environment' of an artifact; that is, the internal structure of a system. The evolution of the system over time is described by the co-evolution of processes, resources, and locations that describe its structure.

Mathematically, we write this as a *transition relation*

$$L, R, E \xrightarrow{a} L', R', E'$$

that is read as the 'the action $a$ occurs using the resources $R$ that are found at location $L$ and leads to the subsequent evolution of the process $E'$ which executes using the resources $R'$ that are found at location $L'$.

For a given model, the definition of this relation is given by a *structural operational semantics* (Plotkin 1981); that is, a system of rules that defines the meaning of each of the combinators. The basic step concerns basic actions and, for this we must specify, for the given model, a specification — which we might call the *signature* of the model — of the effects of the basic actions employed in the model. To do this, we define the *modification function*, $\mu$, which is a partial function that specifies the effect of each basic action on pair of a location and a resource,

$$\{\mu(a, L, R) = (L', R') \mid \text{for basic actions } a\},$$

which is read as 'the effect of executing the action $a$ at location $L$ with available resources $R$ is to move (the computation) to location $L'$ with available resources $R'$. Given this, we can define the basic rule

$$\frac{\mu(a, L, R) = (L', R')}{L, R, a : E \xrightarrow{a} L', R', E}$$

which allows the evolution of a model to begin. For example, an IoT water meter's sensor ($R$) might detect leakage in the basement ($L$) modifying ($a$) the behaviour of the system in order to supress this by closing valves ($R'$) on the ground floor ($L'$).

Now let's consider an IoT system which interacts with another. For example, a building access control system which has an *interface* to the outside world (say a steady stream of people enter the building through the door from outside). To model such interfacing the action $a$ is one that occurs at the boundary of a model, with its environment or another model, then the existence of the action may be determined by sampling a probability distribution (e.g., the distribution of people walking through the door). In this example people's arrival could be represented by a negative exponential distribution that specifies an arrival rate and, when sampled, provides instances of arrivals — that is, actions $a$ — that allow the rule above to fire. This idea relates to Simon's (1996) idea of the 'outer environment', and we shall return to this idea below.

Each of the combinators described above comes along with rule that defines its meaning. For example, the rule

$$\frac{L, R, E \xrightarrow{a} L', R', E' \quad M, S, F \xrightarrow{b} M', S', F'}{N, T, G \xrightarrow{a \times b} N', T', G'}$$

where $N$ denotes the combination of $L$ and $M$, $T$ denotes the combination of $R$ and $S$, and $G$ is the concurrent product of $E$ and $F$.

So, the triples $L, R, E$ describe the state of the system as a point of time in its evolution. A very important question at this point, both mathematically and pragmatically, is when it is that two states, that is two systems, are equivalent. For systems that evolve in the way we have described, the relevant mathematical notion is called *bisimulation*, written

$$L, R, E \sim M, S, F$$

and understood as indicating that, in a rather strong sense, that the two processes, $L$, $R$, $E$ and $M$, $S$, $F$, are able to simulate each other as they evolve.[6]

So far, we have simply described states and their evolution, together with a notion of equality.

$$L, R, E \vDash \phi$$

That is that a model's state $L$, $R$, $E$ satisfies (denoted $\vDash$) the property $\phi$. For example, that the execution of the model (from the current state) terminates, or that a certain amount of resource is consumed/created, or that a given resource is moved from one location to another.

Writing now $\Sigma, \Sigma'$, etc., for system states (i.e., triples $L$, $R$, $E$), we can define this modal logic using a *satisfaction relation* that, case-by-case, gives the conditions under which formulae constructed out of each of the logical connectives in its language is true.

Given this general set, we can describe how the different logical formulae should be understood. First,

$$\Sigma \models \mathrm{p} \quad \text{iff} \quad \Sigma \in V(\mathrm{p}).$$

This is the most basic situation for the system state $\Sigma$. The 'atomic proposition' p is a basic declarative assertion — that is, something to which a truth value can be assigned — that involves no logical connectives. For example, 'the moon is made of blue cheese' or 'the system terminates in finite time'. The assertion p holds for a model $\Sigma$ just in case $\Sigma$ is one of the states that interprets p as being 'true' — to this end, $V$ is a mapping that assigns to each atomic proposition p the set of states in which it is true.

Next come the familiar classical connectives: negation, conjunction, and disjunction. They all work in the evident way:

$$\begin{array}{rcl} \Sigma \models \neg\phi & \text{iff} & \Sigma \not\models \phi \\ \Sigma \models \phi_1 \wedge \phi_2 & \text{iff} & \Sigma \models \phi_1 \text{ and } \Sigma \models \phi_2 \\ \Sigma \models \phi_1 \vee \phi_2 & \text{iff} & \Sigma \models \phi_1 \text{ or } \Sigma \models \phi_2 \end{array}$$

In this setting, classical (material) implication $\phi \supset \psi$ can be defined as $(\neg\phi) \vee \psi$.

A key aspect of the relationship between process-based system models, such as ours, and logic assertions of their properties is described by the so-called 'action modalities'. These come in two basic flavours:

$$\begin{array}{rcl} \Sigma \models \langle a \rangle \phi & \text{iff} & \text{for some } \Sigma \xrightarrow{a} \Sigma',\ \Sigma' \models \phi \\ \Sigma \models [a]\phi & \text{iff} & \text{for all } \Sigma \xrightarrow{a} \Sigma',\ \Sigma' \models \phi \end{array}$$

The first of these is read as 'the formula $\langle a \rangle \phi$ holds for the model $\Sigma$ just in case there is *some* possible evolution of the model $\Sigma$ by the occurrence of the action $a$, defined by the operational semantics as described above, that leads to the new model $\Sigma'$ and the formula $\phi$ holds for $\Sigma'$ '. The second is similar, and is read as 'the formula $[a]\phi$ holds for the model $\Sigma$ just in case for *every* possible evolution of the model $\Sigma$ by the occurrence of the action $a$ that leads to a new model $\Sigma'$, the formula $\phi$ holds for $\Sigma'$.

Finally, we come to some logical formulae that are critical to our account of compositionality in system models. Here $\Sigma_1 \circ \Sigma_2$ denotes the *composition* of the models $\Sigma_1$ and $\Sigma_2$. We have

$$\begin{array}{rcl} \Sigma \models \phi_1 * \phi_2 & \text{iff} & \text{for some decomposition } \Sigma \sim \Sigma_1 \circ \Sigma_2, \\ & & \Sigma_1 \models \phi_1 \text{ and } \Sigma_2 \models \phi_2 \\ \Sigma \models \phi \mathrel{-\!\!*} \psi & \text{iff} & \text{for all } \Sigma' \text{ such that } \Sigma' \models \phi \\ & & \Sigma \circ \Sigma' \models \psi \end{array}$$

When we write $\Sigma \sim \Sigma_1 \circ \Sigma_2$, we are employing the equality on (states) of models called bisimulation that we describe above.

The form of the logical clause for $\phi_1 * \phi_2$ is of particular importance in our setting. It explains how it is that formulae of the form $\phi_1 * \phi_2$ can be used to describe a decomposition of a system into components with

[6] See (Milner 1983; Stirling 2013) for explanations of these ideas.

specified properties. It says that if the model — consisting in locations, resources, and processes, as described above — can be decomposed into two models that are, in an appropriate sense, 'independent', then the properties of the component models can be reasoned about independently. The notion of independence that must be encoded in the composition $\circ$ of models is discussed below, and an example of the decomposition of models is given in Figure 4 (below). One reading of these connectives is in terms of 'resource semantics' (Pym 2019): Focussing on the resources part of the model, the resources required for $\phi * \psi$ are divided into those required for $\phi_1$ and those required for $\phi_2$. The reading of the clause for $\phi -\!\!* \phi$ is similar.

As we have seen, the assertion $L, R, E \vDash \phi$, which lives in the world of *logic*, and is read as 'the state of the system described by the triple $L$, $R$, $E$ has the property $\phi$, expressed in the language of modal logic that has its semantics given in terms of the transition system generated by the operational semantics'.

In general, examples of logical assertions about a system's state include the following:

- the execution of the system terminates after a finite time;
- the amount of the resources of given types used by the system are below a certain upper bound.

In the particular setting of the Internet of Things, examples of logical assertions about a system's state would include things such as

- the water meter is available to give a reading, and
- the water meter's reading has been successfully received by the payments system.

The following theorem — which is attributable variously to van Benthem, Hennessey, and Milner (Anderson and Pym 2016; Collinson et al. 2012; Collinson and Pym 2009; Hennessy and Milner 1980; Milner 1983; Stirling 2013; Van Benthem 2011) — establishes the relationship between the operational semantic and the logical properties of states (recall, writing $\Sigma$ for a state $L$, $R$, $E$):

$$\Sigma \sim \Sigma' \quad \text{iff} \quad \text{for all } \phi,\ \Sigma \models \phi \text{ iff } \Sigma' \models \phi.$$

It says that two states are *bisimilar* — that is, they are operationally equivalent — just in case they satisfy exactly the same logical properties — that is, they are *logically equivalent*.

Simon's (1996) definition refers also to the 'outer environment' of an artifact; that is, the rest of that part of the world within which the system exists. From our modelling perspective, the environment within which a (model of a) system exists can be either a collection of other models of other systems or it can be just something from which events are incident upon the model, as 'inputs' to the model, or to which events from the model are incident, as 'outputs' from the model. As we have explained above, the incidence of events to (and, indeed, from) a model is described using probability distributions.

However, what connects systems to other systems, and to their environments more generally, are *interfaces*. We can think about what we mean by an interface in the setting of our distributed systems models. Caulfield and Pym (2015) have made an initial exploration of this idea in the setting of distributed systems models of the kind we have described. What our treatment of interfaces establishes is a compositional view of distributed systems modelling.[7]

To enable composition, models contain interfaces, which define the locations where models fit together and which actions, defined at appropriate locations within the interface, are party to the composition. Actions in the interface will nevertheless be able to execute only if the resources they require are available.

[7] There are many approaches to systems modelling. We believe that the framework we describe herein is distinguished (i) by its employment of what are, arguably (Anderson and Pym 2016; Collinson et al. 2012; Collinson and Pym 2009; Pym 2019), the minimal semantic structures that are necessary to describe distributed systems, and so the Internet of Things, and (ii) by the tight integration of a logic that supports local reasoning (Ishtiaq and O'hearn 2001; O'Hearn and Pym 1999; Reynolds 2002; Yang and O'Hearn 2002).

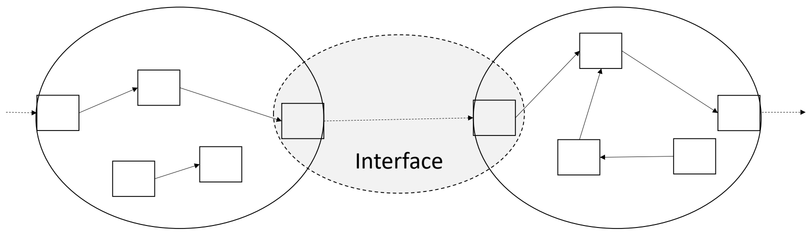


**Figure 3. An interface between two models**

Mathematically, interfaces — depicted informally in Figure 3 — are specified in terms of the mathematical components of system models. The locations and resources of a model are represented using a location graph, $\mathsf{G}(\mathsf{V}[\mathsf{R}], \mathsf{E})$, with a set of vertices, $\mathsf{V}$, representing the locations of the model, and a set of directed edges, $\mathsf{E}$, giving the connections between the locations. Vertices are labelled with resources $\mathsf{R}$.

As explained above, actions evolve the processes and resources of a model. However, rather than thinking of actions evolving processes, it is convenient to think of a process as a trace of actions — the history of actions that have evolved a process during the execution of the model. All of the actions in a model are contained in a set, $\mathsf{A}$, and process traces are comprised of these. The environment a model sits inside causes actions within the model to be executed, at a particular location. A model contains a set of located actions, $\mathsf{L}$, and a located action, $\mathsf{l} \in \mathsf{L}$, is given by an ordered pair $\mathsf{l} = (a \in \mathsf{A}\ , v \in \mathsf{V})$. The environment associates these located actions with *probability distributions*: $Env : \mathsf{A} \rightarrow ProbDist$. During the execution of the model, the located actions are brought into existence by sampling from these distributions.

With this detailed set-up, we can now define the concept of interface for system models. An interface $I \in \mathsf{I}$ on a model is a tuple $(In\ , Out\ , L)$ of sets of input and output vertices, where $In \subseteq \mathsf{V}$ and $Out\,\mathsf{V}$, and a set of located actions $L \subseteq \mathsf{L}$. The sets of input vertices in the interfaces for a model must be disjoint and, similarly, the sets of output vertices in the interfaces for a model must be disjoint.

With this machinery set up, we can define a model

$$M = (\mathsf{G}(\mathsf{V}[\mathsf{R}]\ , \mathsf{E})\ , \mathsf{A}\ , \mathsf{P}\ , \mathsf{L}\ , \mathsf{I}\ )$$

to be a tuple that consists of a location graph, $\mathsf{G}$, a set of actions, $\mathsf{A}$, a set of processes, $\mathsf{P}$, a set of located actions, $\mathsf{L}$, and a set of interfaces, $\mathsf{I}$.

For example, in Figure 4, taken from Caulfield and Pym (2015), we can see two interfaces in action. The figure depicts three models, which together describe the possible information security breaches in a system with employees travelling to and working in an office, composed together by two interfaces. The model in the middle represents the entrance lobby of a building, including a reception desk, security guards, and an access control barrier allowing connection to the inner office, described by the right-hand model. The entrance lobby is accessed from the left-hand model, which describes transport to the building, through a doorway. The access control barrier and doorway are the interfaces between the models.

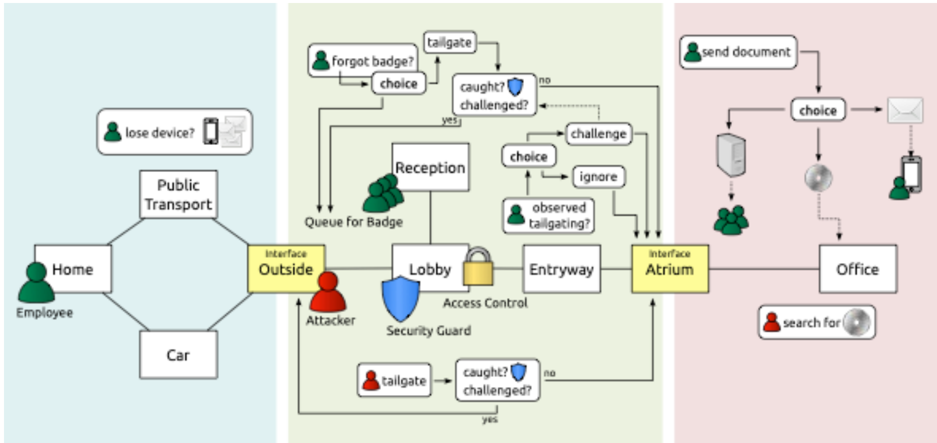


**Figure 4. Composition of models Caulfield and Pym (2015)**

The basic mathematical properties of this concept of interface are explored in Caulfield and Pym (2015).

What's the use of a compositional theory? It manages scale in the design, construction, and understanding of models. In particular, a crucial feature of a compositional theory is the ability to reason independently — that is, locally — about the components of the system in such a way that properties of the composite system can be inferred from those of the components.

## Local reasoning

A key feature of logical systems that support local reasoning of this kind (O'Hearn et al. 2001) — see also (Collinson et al. 2012; Ishtiaq and O'hearn 2001; Pym 2019; Reynolds 2002) for related ideas — are forms of inference called *frame rules* that describe the conditions under which a component of a system can be reasoned about independently of the other components of the system. The following is an example of a frame rule that might apply in our setting:

$$\frac{\{\phi\}\,\Sigma \xrightarrow{a} \Sigma'\,\{\phi'\}}{\{\phi * \psi\}\,\Sigma \xrightarrow{a} \Sigma'\,\{\phi' * \psi\}} \quad a \# \psi$$

This rule, which follows the style of Separation Logic (Ishtiaq and O'hearn 2001; Pym 2019; Reynolds 2002) in using so-called 'Hoare triples' (Hoare 1978), is read as follows:

- the model $\Sigma$ initially has property $\phi$,
- the model $\Sigma$ evolves by the action $a$ to become the model $\Sigma'$,
- the model $\Sigma'$ has property $\phi'$, and
- the action $a$ is *separated* from the property $\psi$, written $a \# \psi$; that is, $\psi$ is a property of $\Sigma$ and $\Sigma'$ that is unaffected by the evolution of $\Sigma$ to $\Sigma'$ by the action $a$.

Critical in the formulation of this rule is the use of the so-called separating conjunction, $*$, in the conclusion of the rule, and the side-condition, $a \# \psi$. The side-condition ensures that $\psi$ describes a property of $\Sigma$ and of $\Sigma'$ that is unaffected by the action $a$. The effectiveness of this rule is best seen by reading the rule upwards from conclusion to premiss, which is how it is typically used in practice, in which case it can be read as follows: if we wish to reason about the action (which may, in general, be a quite complex program) we need to consider only those parts of system and their properties that are directly affected by the action. This is called *local reasoning*.

## Discussion: Modelling of IoT ecosystems within information systems research

We have now described our formal modelling language for *interfaces* and *compositionality*, and described the *frame rules* for local reasoning. This formal modelling language provides a grammar for the construction of extended IoT models based on composition and local reasoning. We now turn our attention to the use of such complete and formally logical models in Information Systems research on IoT.

Such research demands understanding ever more entwined, extended and integrated technical systems (e.g., a SmartCity IoT initiative might involve millions of diverse devices (Botta et al. 2016)). Yet, while our discipline has defined systems in terms which often imply mathematical formalism (e.g., Demetis and Lee (2016) describe; Interrelationships, Holism, Goal-seeking, Transformation, Input–Output, Regulation, Hierarchy, Differentiation, Equifinality (derived from Skyttner (2005, p.53)), or explored formalism within Design Science research, it has limited tools for researching such large-scale IoT initiatives in systemic terms, coupled with limited experience incorporating mathematical formalism into more qualitative and practice-focused studies.

Addressing these gaps, we now show how our formal modelling can construct useful models for undertaking a critical realist study of an IoT enabled 'SmartCities' initiative to provide multimodal transport to citizens of Berlin. Our initiative (loosely based on a case study by (Khannah and Venters 2014) but changed and simplified to explain the use of our modelling), involves using a city-wide travel smartcard to easily move between rented bicycles, trains and, our focus here, rented electric cars. The initiative is constituted from an architecture of locations (e.g., parking space), resources (e.g., cars leased from manufacturers by a leasing company, bicycles, trains, charging points, roads, Apps developed by transport companies, people maintaining and using the service, sensors in cars, parking bays and various other devices) and processes (booking, renting, travelling, payments, charging) identified through qualitative research (interviews and observations). This led to the interpretation that the initiative was divided institutionally (e.g., neighbourhoods, city-travel-initiatives, car-leasing companies and App providers) and technically into systems and interfaces between them (e.g., a booking and payment smartphone App, a charging system, citywide travel systems, car-management system, roads tracking system but also App-Sub-systems etc.).

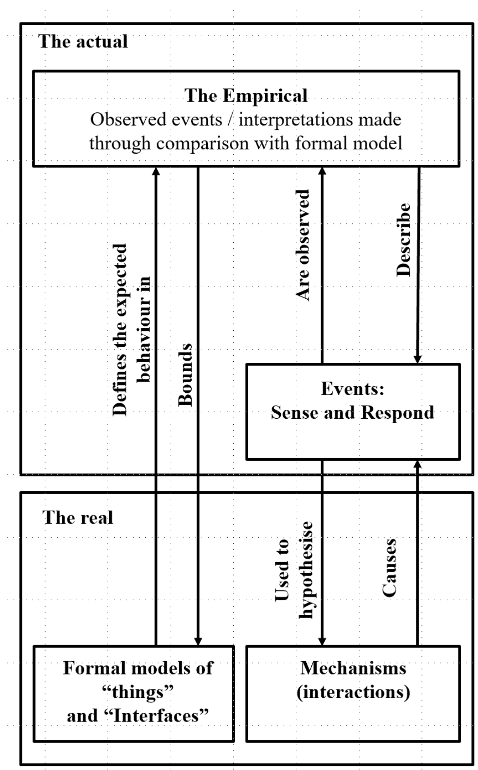


**Figure 5. Our proposed development of CR stratified ontology (based on (Mingers 2004)**

Our complete, systematic language for modelling such compositional systems (a language that is Turing Complete and could thus be used to model any system) can then be applied to model our empirical understanding of the technical elements of this SmartCities initiative.

Considering the role of such modelling within critical realism (see Figure 5), our formal modelling (like systems modelling more generally) exists in the domain of the real by assuming experienced events 'are causally generated by the structures of the underlying system' (Mingers 2000) and using mathematics to define these. Specifically, as logical models they are thus mechanic and real in of themselves but may or may not have any relationship to the reality they attempt to model. Our aim through the modelling process is thus comparison with the empirically observed events observed in the actual (observed events from which the model's boundaries are derived and against which their expressed behaviour are compared). Such comparison is around the expected behaviour shown in our empirical findings compared with the mathematical model and is bounded by the focus of the model (that which it considers).

Our use of modelling thus relates to critical realism's (CR's) retroduction; that is a situation of concern is identified and hypotheses are made through the construction of formal models which describe (hopefully successfully) elements of the empirical observation. Where our work differs from qualitative CR approaches previously used in Information Systems is that our formal mathematical language enables formally complete models of mechanism such that (within the realist belief of CR) any discrepancy from the simulation forms an empirical discrepancy which might lead to identifying new generative mechanisms. In summary our formal models are complete and perfect (mathematically but not necessarily empirically) and thus exist in the domain of the real but are only used to bound and better understand the domain of the empirical (that which is interpreted) from empirical evidence.

Let us now examine a slightly simplified example associated with our Berlin SmartCities initiative. First, we will develop models of the driving and booking, then of the interface between them, and then we will identify potential discrepancies from this model which might lead to further interpretation and the construction of a new generative mechanisms. Note that while we express our example using our mathematical framework, in practice the construction of models is undertaken in a programming language such as Julia, as in (Caulfield and Pym 2015). For illustrative purposes, we give examples of specific resources, locations, and actions rather than generic variables and functions.

Returning to the frame rule:

$$\frac{\{\phi\}\ \Sigma \xrightarrow{a} \Sigma'\ \{\phi'\}}{\{\phi * \psi\}\ \Sigma \xrightarrow{a} \Sigma'\ \{\phi' * \psi\}} \quad a \# \psi$$

- The model $\Sigma$ of a Car Booking initially has property $\phi = \phi_{\text{AlexPlatz},2}$ : that is the Car is at location $L$ = Alexanderplatz and has battery resource is $R$ = 2 hrs of charge.
- The model $\Sigma$ evolves by the action $a$ = *DriveToTegelAirport* to become the model $\Sigma'$.
- The model $\Sigma'$ has property $\phi' = \phi_{\text{TegelAir},h}$ : that is, that the car is at location $L$ = Tegel Airport and has battery resource $R = h$ hrs of charge.
- The action $a$ is *separated* from the property $\psi$, written $a \# \psi$ ; that is, $\psi$ is a property of $\Sigma$ and $\Sigma'$ that is unaffected by the evolution of $\Sigma$ to $\Sigma'$ by the action $a$. Here, we have
  - $a$ is the action that moves a car from Alexanderplatz to Tegel Airport
  - $\psi_{\text{Paid}}$ is a formula that describes the amount for a rental to drive to the airport.
- Then, whether $h = 1$, for a careful driver, or $h = 0$, for an aggressive driver, we can conclude, using the frame rule, that if before the action $a$ occurs the property $\phi_{\text{AlexPlatz},2}$ holds and after the action $a$ the property $\phi_{\text{TegelAir},h}$ holds, then we can conclude also that the property $\phi_{\text{AlexPlatz},2} * \psi_{\text{Paid}}$ holds before the action $a$ occurs and that the property $\phi_{\text{TegelAir},h} * \psi_{\text{Paid}}$ holds after the action $a$. That is, the payment component of the system is independent of the quality of driving.

As things stand, we have described a set-up in which the payment and driving component of this model are separated (the drivers' App handles booking, distance travelled using GPS, and thus billing, whereas the Car's Management Systems handles battery usage).

Further qualitative research may then show a discrepancy. Some drivers drive aggressively, so depleting the battery more than planned and causing problems for the next driver who may not be able to complete their journey. Thus we have revealed a new mechanism (aggressive driving affecting the next journey). We can represent this change through modelling an interface which addresses the aggressive driving mechanism.

In this way, we may wish to model a broader ecosystem in which the amount paid for the rental is affected by the remaining battery charge at the end of the rental period. In this case, the two components would no longer be separated. In particular, we may seek to specify an *interface* between the two components that facilitates this dependency. In terms of our formal definition of interfaces, we should need to introduce in our conceptualization, and hence modelling, of the system, we would need to introduce a logical (i.e., virtual) location, shared between the payment and driving systems, for a sensor device (a resource) that would capture driving data and feed it back, using new system actions located at the shared location, to the payment system, which would itself be modified to perform an alternative calculation.

In terms of our models and their associated logic, we can describe this situation as follows:

- Let $\Sigma$ be a model such as the one sketched above;
- Suppose that $\Sigma$ has two sub models, $\Sigma_1$ and $\Sigma_2$, such as those for payment and driving as described above and suppose that $\Sigma_1 \vDash \theta_1$ and $\Sigma_2 \vDash \theta_2$ are logical judgements that describe the essential properties of the models.
- Then we have that $\Sigma \vDash \theta_1 * \theta_2$;
- Our local reasoning principle (cf. O'Hearn et al. 2001), as sketched above, suggests that we can understand $\Sigma_1$ and $\Sigma_2$ independently. While this is true, a closer examination of the logical structure is quite revealing. Suppose that $\Sigma_1$ and $\Sigma_2$ have an interface, depicted in Figure 3 and defined above. In that case there must be some component of the overall model $\Sigma$ that is shared by the two components $\Sigma_1$ and $\Sigma_2$. In terms of our logic, this can be described by considering $\theta_1$ and $\theta_2$ to share a property, $\chi$, so that we then have

$$\Sigma \models (\phi \wedge \chi) * (\chi \wedge \psi)$$

Unpacking the definition of $*$ given in the definition of $\vDash$, we have

$$\Sigma_1 \models (\phi \wedge \chi) \quad \text{and} \quad \Sigma_2 \models (\chi \wedge \psi)$$

So, according to the definition of $\vDash$ given above, we have

$$\Sigma_1 \models \chi \quad \text{and} \quad \Sigma_2 \models \chi$$

So, the formula $\chi$, which can be understood as the specification of the interface (with the shared sensor location, as described above) between the two components, is satisfied by both components of $\Sigma$. Of course, this explanation employs a very simple account of the logical structure. And of course, much more subtle accounts are possible.

Our next task then is to unpack the minimum formula for $\chi$ that satisfies our desire for *interfacing* the booking App and the Car Management System. At this point the CR researcher might return to their empirical evidence and identify whether their modelling of $\chi$ as a sufficient and satisfactory interface has been realized in practice.

At this point the aim is to examine in the domain of the actual to explore whether the events sensed and responded to (for example the log files of the API usage by the SmartPhoneAPP) fail to meet, meet, or exceed $\chi$. In this way formal modelling is used to create logical structures which, by comparison with our empirical understanding 'explain events that occur (or fail to occur)' through 'hypothesis of underlying mechanisms or systems with causal properties or powers that, through their behaviour and interaction, generate the events we experience' (Mingers 2017)

Such an examination of the Berlin case might then reveal otherwise hidden potential generative mechanisms within the IoT API interface between the App-developers and the Car Management Systems (managed by the leasing company) such as:

- Policy discrepancies presenting security risks — such as encryption decisions reflecting different expectations of risk;
- Misunderstandings of the institutional differences between the developers of these different IoT systems — that is, $\psi$ and $\phi$ — such as using different measurements for 'battery usage' and driving criteria (e.g., speed, acceleration, breaking);
- Structures of power and control — perhaps revealed through whether the Car Management systems calculates who are 'energy-frugal-drivers' or the SmartPhoneApp;
- Exceeding $\chi$ with unnecessary message passing within the interface, perhaps suggesting one party is extracting a data-surplus of financial value. For example, the car-management system might require details of the driver with the aim of selling their driving behaviour data to advertisers or insurers, or the SmartPhone app might require details of the car battery performance — data potentially of value to other battery manufacturers and car manufacturers;

- Failure to provide $\chi$, suggesting limitations on the system and failures of collaboration between the different institutions of the Car-providers (leasers) and App-Developer.

Our aim then would be to examine these discrepancies (through further qualitative research) to derive more general understanding of the interfacing such as *'Car leasing companies are unwilling to share detailed battery characteristics as their business is powerfully influenced by the car manufacturers who wish to keep this data hidden'. A* more general mechanism might then be derived from this discrepancy such as for IoT: *power-asymmetries between manufacturers of the larger structures in which IoT things reside can influence IoT things data-sharing practices.*

We hope that these rather trivial examples illustrate how formal modelling might be used in better understanding the sociotechnical nature of boundary resources (e.g., APIs) within IoT ecosystems.

## Conclusions

This paper provides an account of how to undertake systems modelling cleanly and compositionally and its contribution includes an associated logic for *local reasoning* that enables the partitioning of complex ecosystems into sub-systems and enables the modelling of interfaces between such sub-systems. The paper has subsequently shown, through the application of this logic within IoT ecosystems examples, the potential value of such modelling in practice. In particular, the paper reveals how qualitative and quantitative research of digital ecosystems may be enriched by engagement with the mathematical modelling of key IoT ecosystem components and interfaces and by local reasoning about such IoT devices. By modelling interfaces logically, we are better able to identify discrepancy between expectations and empirical evidence of IoT ecosystems within a critical realist ontology.

More specifically, our research shows how a focus on modelling IoT systems *compositionality* is useful in interrogating interfaces such as APIs within an IoT ecosystems. Yet modelling of separation and compositionality are mostly absent from our existing modelling approaches (e.g., BPM, IDEF0, ArchiMate and UML) and, where they are included, they are informally defined with interfaces assumed unproblematic. Indeed, most modelling approaches appear founded upon the assumption of a single administrative domain (e.g., an organization) in control of the entire system rather than an ecosystem of different administrative domains. In contrast, our compositionality is organized using a modern logic that is useful for *local reasoning* about IoT sub-systems and their relationship with other sub-systems through *compositionality*. It thus addresses the call for the modelling of systems that are 'more than the sum of [their] parts' (Demetis and Lee 2016) by providing a rigorous way to model multiple isolated and interfaced technical parts and identify what the sum is expected to be, and what is missing from such a sum (including social features). In this way the paper directly addresses the call for research on boundaries and components within widely distributed networks (Yoo et al. 2010).

Our modelling is placed within a critical realist ontology because of the need to acknowledge that such detailed modelling is unfeasible for researching most large ecosystems and that such ecosystems are socio-technical rather than purely technical — meaning they are inherently open and interactive. Rather we suggest the use of fragments of modelling (undertaken through software) within explanatory critiques (Bhaskar 2009) that re-unite facts (represented in such model software) with values (represented in the comparison with evidence of the world and the social structures within it). Our aim is to isolate expected and unexpected regularities (e.g., a policy decision within the choices taken over an IoT interface). Our research occurs in the domain of the actual (e.g., we observe what happens) and by comparison with our logical model in the domain of the real we are able to identify discrepancy: *'The empiricist identification of causal laws with empirical regularities thus involves a double reduction — that of laws to events and events to experiences.'* (Mingers 2000).

Ultimately, our aim is to avoid a naïve realism in which we would suggest that mathematical formalism is valid for modelling of entire socio-technical IoT ecosystems as this would be astronomically expensive, infeasible, and always partial (assuming stability and regularity absent in IoT reality and denying the agency of social factors). Our ontology also recognises the 'cultural and temporal limitations of our access to that world' within the domain of the empirical (Mingers 2017). Remembering that the domain of the real is a 'complex interaction between dynamic, open, stratified systems, both material and non-material, where particular structures give rise to certain casual powers, tendencies, or ways of acting' (Mingers 2000), our aim instead is to use formal modelling to provide rigorous structures against which to compare empirical findings (e.g., of IoT devices), so supporting the devising of relevant 'hypothetical mechanisms which, if they existed, would generate or cause that which is to be explained' (Mingers 2000). The next step would be to evaluate these potential generative mechanisms against the formal logical modelling in order to identify currently correct generative mechanisms. Our aim, then, is to explore divergence from our models — which remain partial accounts located in the domain of the real — but which, through compositionality and local reasoning, allow a focus on interfaces within IoT ecosystems.

Further research is needed that deeply examines IoT ecosystems, drawing on detailed empirical evidence (qualitative and quantitative) to develop parametrized models. Such models could then be used to evaluate the use of our modelling within studies of digital ecosystems in general and IoT in particular. Further research is also needed to integrate better mathematical modelling within the study of digital ecosystems and, in particular, theorizing interfaces within our socio-technical understanding of such ecosystems.

## Acknowledgements

This research was partially supported by the UK's Engineering and Physical Sciences Research Council (Grant EP/R006865/1).

## References

Alter, S. 2004. "Desperately seeking systems thinking in the information systems discipline," *ICIS 2004 proceedings*), p. 61.

Anderson, G., and Pym, D. 2016. "A Calculus and Logic of Resources and Processes.," *Theoretical Computer Science* (614:C).

Atzori, L., Iera, A., and Morabito, G. 2010. "The Internet of Things: A survey," *Computer Networks* (54:15), pp. 2787-2805.

Barrett, M., Davidson, E., Prabhu, J., and Vargo, S. L. 2015. "Service innovation in the digital age: Key contributions and future directions," *MIS Quarterly* (39:1), pp. 135-154.

Bhaskar, R. 2009. *Scientific realism and human emancipation*. Routledge.

Botta, A., de Donato, W., Persico, V., and Pescapé, A. 2016. "Integration of Cloud computing and Internet of Things: A survey," *Future Generation Computer Systems* (56), pp. 684-700.

Caulfield, T., and Pym, D. 2015. "Modelling and simulating systems security policy," *SIMUTOOLS 2015-8th EAI International Conference on Simulation Tools and Techniques*: ICST.

Checkland, P. 1981. *Systems Thinking, Systems Practice*. Chichester: Wiley.

Checkland, P., and Holwell, S. 1998. *Information, Systems and Information Systems*. Chichester: John Wiley & Sons.

Collinson, M., Monahan, B., and Pym, D. 2012. *A discipline of mathematical systems modelling*. College Publications.

Collinson, M., and Pym, D. 2009. "Algebra and logic for resource-based systems modelling," *Mathematical Structures in Computer Science* (19:5), pp. 959-1027.

Constantinides, P., Henfridsson, O., and Parker, G. G. 2018. "Introduction—Platforms and Infrastructures in the Digital Age," *Information Systems Research* (29:2), pp. 381-400.

Coulouris, G., Dollimore, J., Kindberg, T., and Blair, G. 2011. *Distributed Systems: Concepts and Design*, (5th ed.). Pearson.

Demetis, D. S., and Lee, A. S. 2016. "Crafting theory to satisfy the requirements of systems science," *Information and Organization* (26:4), pp. 116-126.

Ghazawneh, A., and Henfridsson, O. 2013. "Balancing platform control and external contribution in third-party development: the boundary resources model," *Information Systems Journal* (23:2), pp. 173-192.

Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. 2013. "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future generation computer systems* (29:7), pp. 1645-1660.

Henfridsson, O., and Bygstad, B. 2013. "The Generative Mechanisms of Digital Infrastructure Evolution," *MIS Quarterly* (37:3), pp. 896-931.

Hennessy, M., and Milner, R. 1980. "On observing nondeterminism and concurrency," *International Colloquium on Automata, Languages, and Programming*: Springer, pp. 299-309.

Hoare, C. A. R. 1978. "Proof of correctness of data representations," in *Programming methodology*. Springer, pp. 269-281.
Ishtiaq, S. S., and O'hearn, P. W. 2001. "BI as an assertion language for mutable data structures," *Proceedings of the 28th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pp. 14-26.
Kallinikos, J. 2004. "Farewell to constructivism: technology and context-embedded action," in *The Social Study of Information and Communication Technology: Innovation, Actors and Contexts,* C. Avgerou, C. Ciborra and F. Land (eds.). Oxford: Oxford University Press.
Karhu, K., Gustafsson, R., and Lyytinen, K. 2018. "Exploiting and Defending Open Digital Platforms with Boundary Resources: Android's Five Platform Forks," *Information Systems Research* (29:2), pp. 479-497.
Khannah, A., and Venters, W. 2014. "Exploring the rhythms of information infrastructure coordination for smart cities: The case of building a mobility infrastructure in Berlin.," *22st European Conference on Information Systems,* C. Avgerou (ed.), Tel-Aviv, Israel.
Mathiassen, L., and Sørensen, C. 2008. "Towards A Theory of Organizational Information Services," *Journal of Information Technology* (23:4), pp. 313–329.
Milner, R. 1983. "Calculi for synchrony and asynchrony," *Theoretical computer science* (25:3), pp. 267-310.
Mingers, J. 2000. "The Contribution of Critical Realism as an Underpinning Philosophy for OR/MS and Systems," *Journal of Operations Research Society* (51:11), pp. 1256-1270.
Mingers, J. 2004. "Real-izing information systems: critical realism as an underpinning philosophy for information systems," *Information and Organization* (14:2), pp. 87-103.
Mingers, J. 2011. "The Truth, The Whole Truth, and Nothing but the Truth.," in *The Oxford Handbook of Management Information Systems,* B. Galliers and W. Currie (eds.). Oxford: Oxford University Press, pp. 67-86.
Mingers, J. 2017. "Back to the future: A critique of Demetis and Lee's "Crafting theory to satisfy the requirements of systems science"," *Information and Organization* (27:1), pp. 67-71.
Nicolescu, R., Huth, M., Radanliev, P., and De Roure, D. 2018. "Mapping the values of IoT," *Journal of Information Technology* (33:4), pp. 345-360.
O'Hearn, P. 2019. "Separation logic," *Communications of the ACM* (62:2), pp. 86-95.
O'Hearn, P. W., and Pym, D. J. 1999. "The logic of bunched implications," *Bulletin of Symbolic Logic* (5:2), pp. 215-244.
O'Hearn, P., Reynolds, J., and Yang, H. 2001. "Local reasoning about programs that alter data structures," *International Workshop on Computer Science Logic*: Springer, pp. 1-19.
Orlikowski, W. J., and Iacono, C. S. 2001. "Research Commentary: Desperately Seeking the "IT" in IT Research--A Call to Theorizing the IT Artifact," *Information Systems Research* (12:2), pp. 121 -134.
Pickering, A. 2010. *The Cybernetic Brain*. Chicago: Chicago University Press.
Plotkin, G. D. 1981. "A structural approach to operational semantics,").
Pym, D. 2019. "Resource semantics: logic as a modelling technology," *ACM SIGLOG News* (6:2), pp. 5-41.
Reynolds, J. C. 2002. "Separation logic: A logic for shared mutable data structures," *Proceedings 17th Annual IEEE Symposium on Logic in Computer Science*: IEEE, pp. 55-74.
Simon, H. A. 1996. *The sciences of the artificial*, (third edition ed.). Cambridge, MA: MIT Press.
Skyttner, L. 2005. *General systems theory: Problems, perspectives, practice*. World scientific.
Stirling, C. 2013. *Modal and temporal properties of processes*. Springer Science & Business Media.
Tiwana, A., Konsynski, B., and Bush, A. A. 2010. "Research Commentary—Platform Evolution: Coevolution of Platform Architecture, Governance, and Environmental Dynamics," *Information Systems Research* (21:4), pp. 675-687.
Van Benthem, J. 2011. *Logical dynamics of information and interaction*. Cambridge University Press.
Yang, H., and O'Hearn, P. 2002. "A semantic basis for local reasoning," *International Conference on Foundations of Software Science and Computation Structures*: Springer, pp. 402-416.
Yoo, Y., Henfridsson, O., and Lyytinen, K. 2010. "The New Organizing Logic of Digital Innovation: An Agenda for Information Systems Research," *Information Systems Research* (21:4), pp. 724-735.