Macro-types in multi-scale feedback systems: a conceptual framework

Ada Diaconescu
LTCI, Telecom Paris, FR
diacones-at-telecom-paris.fr

Louisa Jane Di Felice
Universitat de Barcelona, ES
louisajane.df-at-gmail.com

Patricia Mellodge
University of Hartford, USA
mellodge-at-hartford.edu

Payam Zahadat
IT University of Copenhagen,DK
paza-at-itu.dk

***Abstract*—Large Complex Adaptive Systems (CAS) operate at multiple scales – processing various information flows, encoded at different granularities, over different substrates. Often, system scales are interconnected by feedback cycles, with micro-scales generating macro-scales, in turn influencing the micro-scales, and so on. E.g. foraging ants (micro) contribute to a pheromone trail (macro), which in turn guides their movement (micro) and updates the trail (macro). As these processes take widely different forms when occurring in different environments and accomplishing different functions, they may be difficult to detect, compare and transfer across CAS domains. We previously generalised such diverse processes, defining them explicitly as a Multi-Scale Feedback Systems (MSFS) design pattern. In this paper, we propose a conceptual framework for categorising macro-processes in MSFS. This helps identify and analyse micro-macro feedback cycles across CAS domains; and engineer new CAS with specific constraints. We illustrate these concepts via CAS examples that fit the MSFS pattern via different variants. This contribution extends the theoretical basis of MSFS for dealing with the increasing complexity of modern environments.**

***Index Terms*—Macro, micro, multi-scale, feedback, framework, deployment types, design pattern, complex adaptive systems**

I. Introduction

Large Complex Adaptive Systems (CAS) consist of numerous interacting components, typically giving rise to multi-scale phenomena, where micro-interactions lead to macro formations [1], [2], [3]. Multi-scale systems take many forms, including self-encapsulating structures (e.g. organisms made of cells and of atoms); collective actions (e.g. ants foraging via a shared pheromone trail); or hierarchical organisations (e.g. reporting and decisions in human enterprises). Often, such multi-scale phenomena are interconnected by feedback loops, where macro processes feedback onto micro processes, causing them to adapt (i.e. downward causation [4], [5]) and in turn change the macro phenomena [6], [7], [8].

Multi-scale concepts have been adopted by scientists for modelling and studying existing CAS; and by engineers for developing new CAS. Common multi-scale system designs include either multi-level (or layered) architectures [9]; or hierarchies, featuring a tree-like topology and top-down control authority [10], [11], [12]. Here, macro-entities are typically centralised, symbolic, external representations of micro entities. Alternatively, several designs also consider decentralised macro representations – e.g. bio-inspired design patterns such as reaction-diffusion or gossiping [13]; aggregate computing [14]; or swarm systems [15]. Notably, these were mostly limited to two scales, generating a single micro-macro feedback.

We previously generalised such multi-scale phenomena via a generic design pattern, Multi-Scale Feedback Systems (MSFS) [16], where micro-macro feedback loops are integrated recursively, forming multiple quasi-independent scales that adapt to each other. We also highlighted design-driven trade-offs between scale stability and adaptability, – e.g. [17] showed the effect of inter-scale abstraction parameters and delays on system behaviour and attractors.

Support for multiple scales allows addressing increasing system size and complexity. Each scale in MSFS is a macro-entity with respect to micro-entities at the underlying scale that generated it – i.e. it is an abstract representation of the collective state of the micro-entities. The abstract representation (macro-scale) is a description at a coarser information grain relative to a detailed description of each micro-entity and each interaction (micro-scale). To limit required coordination resources, micro-entities can perceive their collective state via the abstract representation, at the macro-scale, rather than via a detailed representation at the micro-scale. They can adapt accordingly forming a micro-macro feedback loop.

This approach helps to manage complexity in large systems by relying on recursive simplifications of system representations (e.g. models). Typically, higher scales represent larger system parts, or scopes, in less detail relative to lower scales that cover narrower scopes in more detail. This does not mean that higher scales are less complex than lower ones, as they may include extra information sources that do not concern lower scales (e.g. contextual learning and decisions). However, higher scales do contain less information than they would if they were to describe the entire low-scale system in detail.

While we consider MSFS as an essential conceptual pattern for understanding or developing large CAS, we notice the preponderance in existing solutions of hierarchical, multi-layer or two-scale decentralised designs. Large natural CAS indicate that the viable design space is much larger, with variants featuring diverse qualitative properties (e.g. robustness vs. performance). Hence, focusing on a narrow subset of variants (e.g., prioritising performance, as in top-down hierarchies) risks missing more suitable design opportunities for CAS with different requirements (e.g. robustness, resilience).

Our key contribution is a novel conceptual framework for: i) more thoroughly defining micro-macro abstractions and feedback; which then helps ii) identifying and categorising macro-phenomena that may occur in a wide variety of forms and representations. We aim to go beyond common MSFS designs, to include less intuitive macro-entity forms, e.g., ones that may occur onto mobile substrates; or be included within micro-entities; or encoded structurally (substrate-dependent) or symbolically (substrate-transferable) [18], [19].

We neither claim nor aim to provide a "correct" or "optimal" conceptual framework. Like with most complex concepts (e.g. complexity, intelligence, life) the borders between conceptual dimensions may become fuzzy in some cases, and may need to be extended, refined or adapted to specific application domains and environments. We consider proposed macro-types as shareable intuitions, or collective agreements, about groups of macro-information instances with similar properties.

The proposed contribution aims to extend the concept of macro-phenomena so that it can be recognised within a wider range of occurrences. This can help scientists and analysts of existing CAS to detect various forms of macro phenomena; and system engineers to rely on a broader design range for CAS, to meet application-specific constraints.

II. Case studies

First, we provide a basic MSFS example for illustration purposes, to help render conceptual notions more concrete. Then, we employ three examples from our previous studies [20] to show how the proposed conceptual framework (sec. V) can be employed to analyse various macro-phenomena (sec. VI). The case studies are summarised below.

A. Illustrative example for Collective Averaging (CA)

This simple MSFS (Fig. 1) consists of 4 micro-entities with initial states: 1, 2, 3, 4. They aim to coordinate to reach the same state – e.g. their average: 2.5. Each micro-entity can adapt by adding or removing $\Delta = 0.1$ to their state, to approach the collective state of the others. To estimate their collective state, they can communicate with each other directly or via other entities. All entities have limited resources, only allowing operations between two numbers. Hence, they can organise into a three-scale MSFS (Fig. 1): two mid-entities average two micro-entities each; and a top-entity averages the two mid-entities. The top average is sent as feedback to micro-entities, which add or remove Δ to their state to approach their perceived average. This forms a three-scale micro-macro cycle that takes one step; the MSFS converges (to 2.5) in 15 steps.

While basic, this example corresponds to many equivalent phenomena across diverse domains (e.g. related to entropy maximisation; or collective synchronisation– see the Hierarchical Oscillators example below).

B. Our theoretical case studies

1) Hierarchical Oscillators (HO): a set of four chemical oscillators, with substance concentrations varying in time, aim to synchronise and feature the same concentrations at the same time – similar to the CA example (II-A). They organise into a three-scale MSFS: two mid-entities average concentrations

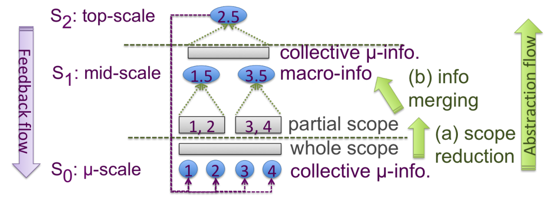

Fig. 1. Example of three-scale feedback system for collective averaging (CA)– performing micro-macro information abstraction in two-steps: (a) scope-reduction (subset); and (b) information-merging (average)

of two oscillators each; and a top-entity averages the two mid-aggregators' averages. Unlike CA, feedback from the top is sent back progressively, through the mid-entities, to the oscillators. These latter adapt to approach the average concentration of their mid-aggregator (cf. details in [16], [17]).

2) Robot Collective (RC): a set of mobile robots aim to distribute across four rooms (grid layout) to best match the number of items in each room. Robots observe the room locations of items and other robots they encounter, as well as the other robots' aggregate observations. Internally, robots represent collected information at three scales: sensory information (micro-scale); estimated robot&item counts (mid-scale); and goal room (top-scale). The top scale provides feedback to the robots (micro-scale), which move towards the designated room (cf. details in [20]).

3) Collective Decisions (CD): a set of agents aims to manage resources of two types in an environment, to ensure sustainability. They need to collectively estimate the ongoing resource distribution between the two types: if they over-estimate a type, that resource type will decrease over time; and if they sub-estimate it, the resource will increase over time. The environment is sustainable if neither type disappears or increases out of control. Typed resources (micro-scale) can change their types based on feedback from the agents. Each individual agent (mid-scale) samples the resource types in its vicinity; and forms an individual opinion O_i. Agents then discuss their respective opinions O_i and reach a consensus O_{col}. This leads to a policy document (top-scale) engendering environmental action. This latter provides (indirect) feedback to resources, which change their types accordingly.

III. Background and previous work

A. Information systems and substrates

In any system, component interactions imply some sort of exchange of matter, energy or information. When material or energy flows are perceived by a component, i.e. an *agent observer*, which consequentially changes its knowledge or behaviour, we conceptualise these flows as *information flows*. An observer's perception does not imply consciousness, only the ability to adapt to perceived information [21]. E.g. observers may be organisms, cells, software or mechanical systems. In this sense, all material and energy flows are potentially informational, depending on their observers. The same applies to all parts of a CAS (cf. informational structural realism [22]).

Still, information always requires a physical substrate [5], to which it can be more or less coupled. E.g. gas pressure (info) is strongly coupled to molecules (substrate); yet pressure recordings (info) onto a sensor screen or a notebook (substrates) only loosely depend on these substrates (cf. structural vs. symbolic information [18], [19]). Extracting information from a substrate requires energy (cf. 'negentropy' [23]).

B. Information scales and micro-macro feedback cycles

An information *scale* is a granularity of observation for a targeted object [24]. This means that an object's scale is the granularity at which it is perceived by an observer. This may differ from the granularity at which the observer represents the object, based on its perception. Scales can be spatial, temporal, or organisational – this offers a unifying concept for several equivalent terms, e.g. abstraction level, layer [9] or lens [25]; blurred or coarse-grained [4] observations.

In CAS, interacting entities (micro-scale) may lead to 'emergent' phenomena (macro-scale), that are simpler to describe overall than all components and interactions [8], [26]. Hence, an observer of the emergent macro-entity would perceive it at a coarser granularity relative to the micro-entities that generated it. In some cases, a macro-entity can feedback onto the micro-entities, which adapt in consequence – cf. downward causation [4]. This leads to the formation of a *micro-macro feedback cycle*, where the micro-scale generates the macro-scale, which represents an abstraction of the collective micro-states, and in turn modifies the micro-scale. We can consider that the macro-entity observes the micro-entities at a lower granularity (micro-scale) than its own representation (macro-scale); and in turn, the micro-entities observe the macro-entity at a coarser granularity (macro-scale) than their representations (micro-scale). Such micro-macro feedback cycle may also occur when a macro-entity actively observes interacting micro-entities and forms a model (macro-information, or abstract representation) of these micro-entities; and consequently acts upon them, leading to model changes and further action on micro-entities.

The key aspects of a micro-macro feedback cycle are: i) a macro-entity encodes macro-information that is an abstraction of micro-information encoded by micro-entities, i.e., the macro-entity contains less information than the collective of micro-entities; and, ii) macro-entities feed macro-information back onto micro-entities, which adapt accordingly.

The above process may repeat recursively at increasing scales, as macro-scales become micro-scales for higher macro-scales, thus leading to *multi-scale feedback cycles* [16], [17]. Previously, we formalised these as a CAS design pattern, MSFS [7], [16]; analysed its sensitivity to inter-scale abstraction parameters [27], time delays [17] and adaptation strategies [16]; and proposed specific measurements for their information flows [20].

C. Previous macro-entity categories

In previous work, we identified different types of macro-information, mostly considering the macro-substrate's relation to the micro-substrate [7]: i) *exogenous*: macro-substrate external to the micro-substrate; ii) *micro-distributed*: macro-substrate internal to the micro-entities (each micro-entity forming a macro-representation of other micro-entities); and iii) *composite*: macro-substrate made-of the micro-entities (macro-information consisting of the micro-substrate's structure).

These categories suffice for distinguishing between frequent macro-entity forms, while unifying related vocabulary from the literature. Roughly, hierarchies and multi-level architectures involve exogenous macro-types; decentralised collectives with shared rules or norms correspond to micro-distributed macro-types; and holonic, self-encapsulated designs in inert matter or organisms feature composite macro-types [1].

Yet, these categories fail to distinguish between more subtle macro categories, going beyond substrate differences. A distributed macro-entity can also be exogenous to the micro-entities – e.g. the set of individual agent models O_i relative to observed resources in the CD case (II-B3). Or, both exogenous and endogenous macro-entities can be encoded via symbolic or structural representations – e.g. structural substance concentration endogenous to oscillators (HO case, II-B1), or structural concentration of a pheromone path exogenous to foraging ants.

To address these limitations, we further distinguished in [20] between *centralised* and *distributed* macro-types (depending on their substrate's spatial coherence and flexibility); and *symbolic* and *structural* macro-types (depending on their information encoding). However, these categories still cannot distinguish between various degrees of distribution, flexibility and perception scope of macro-entities – e.g. there is no difference between: macro-substrates that are statically or dynamically networked, or mobile; complete or partial macro-models of micro-entities; uniform or heterogeneous perceptions of macro-entities by similar observers. Such macro-type variants may significantly impact the performance or robustness of the resulting MSFS.

Via the current proposal, we aim to unify, reorganise and extend previous work, to explicitly focus on the macro-type issue and help identify its various forms more rigorously and extensively. These forms may be further refined and extended depending on each application and its constraints.

IV. Conceptual micro-macro framework

To characterise macro-types, we must first get a better understanding of micro- and macro-entities, observers, objects and scales; and their relations. We can then rely on these concepts to differentiate macro-types (sec. V).

A. Macro-types, observers and objects

A **macro-type** is a categorisation of macro-information according to several criteria, or comparison dimensions. Macro-information is an abstraction of information from multiple sources (in general, though it could be from a single source), which we generally refer to as collective micro-information. In practice, a macro-entity may merge information from micro-entities, but also from other entities at the same or higher representation scales; or from the environment.

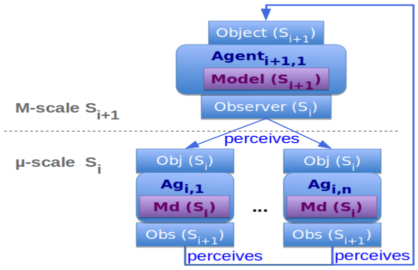

Fig. 2. An Agent (Ag_{S_i}) plays a double role: Observer (Obs) and Object (Obj). It specifies collected information via a Model (Md) at scale S_i. It collects information from observed objects at lower scale S_{i-1} and/or higher scales S_{i+1}. It exposes information at its scale S_i to other agents.

We refer to observed macro-information as a **macro-object**, to emphasise its role as an observation target. A macro-object is a representation of the collective states of micro-entities from a lower observation scale. A macro-object's macro-type critically depends on an **observer** that perceives it– the macro-type may change when perceived by a different observer.

B. Agents, models and scales

Agents are MSFS entities that perceive, process, store and provide information. Externally, they play both the roles of observers and objects (subsec. IV-E). An agent contains an internal **model**, that represents an abstraction of its state, other agents' states and the environment (Fig. 2). An agent at scale S_s: i) collects information from agents at different scales (S_{s-k}, $k{>}0$ from the abstraction flow, and S_{s+k} from the feedback flow); ii) updates its model at scale S_s; and iii) observers its model to decide, act and inform other agents.

An agent's scale is given by the representation granularity of its model. This is the scale at which the agent is perceived, by itself or others. It is different from the scale at which the agent perceives other objects from different scales. Namely, an agent at scale S_s that perceives a bottom-up information flow (abstraction), represents its model at a granularity (S_s) that is *higher* than the model granularity of perceived objects (S_{s-k}, $k{>}0$). Conversely, an agent at scale S_s that perceives a top-down flow (feedback) represents its model at a granularity (S_s) that is *lower* than that of perceived objects (S_{s+k}, $k{>}0$). In brief, an agent's (representation) scale is *higher* than its observation scale for the abstraction flow; and *lower* than its observation scale for the feedback flow.

Agents typically have a goal, which they pursue via information collection, processing and adaption. They may or may not be aware of the goal, which may be encoded explicitly as stand-alone information or included implicitly in the agent behaviours. In a MSFS, some agents may hold an explicit goal, and influence via feedback the other agents to adapt collectively towards the goal. In our case-studies here, goals are implicit in the agents' behaviours; we also studied MSFS case studies featuring explicit agent goals [16], [28].

C. Inter-scale information abstraction

Micro-macro abstraction intervenes when information about the collective state of entities at a lower scale is summarised into a shorter representation at a higher scale. It implies information loss between scales – achievable in two ways: (a) *scope-reduction*: diminishing the information domain; (b) *information-merging* ('coarse-graining'): increasing the granularity of information representation. As MSFS abstraction usually implies both, we only refer to information granularity by default, when qualifying scales (while implicitly also supposing scope-reduction; unless mentioned otherwise).

The information *scope* defines the set of micro-information items representing the collective micro-state. It can be spatial (e.g. physical area or virtual domain from which micro-information is sampled), or temporal (e.g. a sampling rate or gliding window). *Scope-reduction* of the collective micro-state results in a partial scope, with less information, at the same granularity. *Information-merging* aggregates micro-information into macro-information with coarser granularity.

We illustrate these concepts via the collective averaging example (II-A), Fig. 1. Consider the abstraction of micro-information $\{1, 2, 3, 4\}$ (collective micro-state) to macro-information $\{1.5\}$ (the average model of the mid-entity on the left). This abstraction involves two steps: (a) scope-reduction of the collective micro-state $\{1, 2, 3, 4\}$ to a partial scope: $\{1, 2\}$; (b) information-merging from the partial scope into its average: $\{1.5\}$ (for the left mid-entity). We note that this mid-average, $\{1.5\}$, contains less information than the original micro-set $\{1, 2, 3, 4\}$. The abstraction from the mid- to the top-scale is similar, yet without scope-reduction. Micro-entities receive the resulting top-average $\{2.5\}$, adapt their states to $\{1.1, 2.1, 2.9, 3.9\}$ to approach it, and the cycle restarts.

D. Types of micro-macro observers and objects

We distinguish two types of micro-objects (Fig. 3):

- *Individual micro-object* (μ-obj): the perceivable state of each micro-entity (e.g. $\{1\}$, $\{2\}$, $\{3\}$, $\{4\}$, separately);
- *Collective micro-object* (μ_{Col}-obj): the set of all micro-objects over the whole micro-scope (e.g. $\{1, 2, 3, 4\}$).

These micro-objects provide information sources for two types of macro-objects, corresponding to the two abstraction ways:

- *Macro-collective object* (M_{Col}-obj): the part of μ_{Col}-obj obtained via scope-reduction (a) (e.g. $\{1, 2\}$);
- *Macro-model object* (M_{Mod}-obj): the macro-information obtained after information-merging (b) (e.g. $\{1.5\}$). This becomes a micro-object (μ-obj) for the next scale.

All objects represent the perceivable states of entities within an observation scope, relative to an observer. This leads to four observer types, one for each object type. As we focus on macro-types, we only consider two macro-observers:

- *Macro-collective observer* (M_{Col}-obs): perceives a partial-scope of the collective micro-state (M_{Col}-obj);
- *Macro-model observer* (M_{Mod}-obs): perceives the macro-model object (M_{Mod}-obj).

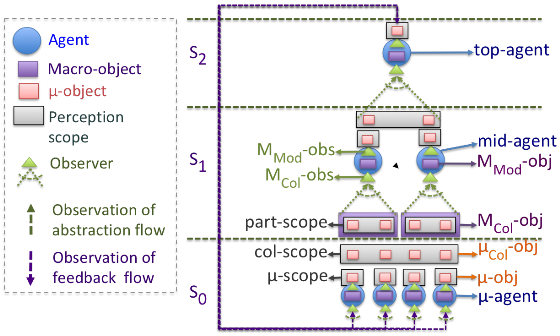

Fig. 3. Micro and macro types of observers and objects

These macro-observers are present in each agent. M_{Col}-obs performs the two micro-macro abstractions: (a) scope-reduction, by only perceiving part of the μ_{Col}-obj (i.e. M_{Col}-obj, at micro-scale) – via the abstraction flow; it may also perceive other M_{Col}-obj from higher scales – via the feedback flow; and (b) information-merging, by aggregating micro-information perceived from M_{Col}-obj into a coarser-grained representation (i.e. M_{Mod}-obj, at macro-scale). M_{Mod}-obs perceives the model (i.e. M_{Mod}-obj) and uses it for adaptation or as a μ-obj for other observers, at different scales. In principle, the M_{Mod}-obs may also collect other information and provide a further abstraction as μ-obj for external observers; yet we do not focus on this here. As a reference, we also define a hypothetical *external observer* (Ext-Obs), able to perceive the entire collective micro-state (μ_{Col}-obj) – e.g. a scientist.

In summary, we have micro-objects (μ-obj) forming a collective (μ_{Col}-obj). An agent perceives part of this micro-collective, via scope-reduction. We refer to this agent observer as macro-collective observer (M_{Col}-obs) and to the partial view of the micro-collective as macro-collective object (M_{Col}-obj). M_{Col}-obs further abstracts the M_{Col}-obj, via info-merging, forming a macro-model object (M_{Mod}-obj). Another agent observer employs this model (M_{Mod}-obj) to take decisions, act, and inform other agents. We refer to this observer as the macro-model observer (M_{Mod}-obs).

E. Agent roles and attributes

Externally, an agent plays a double role (Fig. 4): i) *macro-observer* (M_{Col}-obs) of partial micro-object sets (M_{Col}-obj); and ii) *micro-object* (μ-obj) as part of a set perceived by macro-observers at another scale. Macro-observer roles can be played actively (e.g. a robot collecting and modelling the collective states of other robots); or passively (e.g. a pheromone trail that receives molecule deposits from ants).

Fig. 4 highlights the main attributes characterising agents, observers and objects (focusing on macro-observers and macro-objects). These attributes constitute the basis for defining macro-types (sec. V).

Objects are represented at a *scale* (that is the same as that of the agents they represent) and extend over a *scope* (representation domain). Observers are characterised by a *sampling method* for extracting information from the perceived

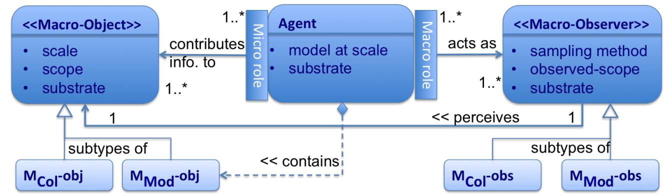

Fig. 4. Overview of the main conceptual entities: agents playing the double role of contributor (micro-role) and observer (macro-role) of macro-object(s)

macro-object – at the granularity given by the macro-object's representation and within the domain given by the macro-object's scope. Sampled observations are within an *observed-scope*, which may be the same or smaller than the macro-object's scope (e.g. if a mid-manager in Fig. 1 only picked one of two numbers in the partial observation scope).

All agents, observers and objects are encoded onto a *substrate*, which they may or not share.

V. Comparison dimensions for macro-types

A. Overview

We identify several dimensions that we consider relevant for comparing different aspects of macro-entities (Table I). All combinations of dimension values may be valid, yet analysing the entire type space here is out of scope. All dimensions concern a macro-entity, but only relative to micro-entities that lead to it; and only as perceived by a specific observer.

Most dimensions characterise the substrate onto which information is encoded: does the macro-object rely on the same substrate as its observer and micro-entities, or on different substrates? Is the macro-object's substrate monolithic, rigid and static; or is it scattered, flexible and dynamic? We also analyse the relation between the macro-information and its substrate: is the macro-object highly dependent on its substrate or can it be easily translated onto different substrates? Finally, we consider two aspects that characterise the actual observations, i.e., the observer's relation to the macro-object: are observations comprehending the entire macro-phenomena or only parts of it? And in the latter case, are these parts co-localised or dispersed? Will multiple observations from similar observers be identical, equivalent or dissimilar enough to ensue different consequences for the observer?

B. Relative substrates

The first set of macro-type dimensions relate to the substrate upon which the macro-information is encoded. This macro-information substrate (or macro-substrate) can be categorised relative to the substrate of: i) the micro-entities that generate the macro-object, i.e. **substrate relative to micro**; and ii) the observer perceiving the macro-object, i.e. **substrate relative to observer**. In both cases, the macro-substrate can be either *exogenous* or *endogenous* with respect to the substrate of the relative entity (micro-entities or macro-observer). Exogenous means that the macro-substrate is external or different to the substrate of the relative entity; and endogenous means that it is internal to the relative entity or based on the same substrate.

TABLE I
MACRO-TYPES FOR MSFS

Macro dimension	Types	Examples
Substrate relative to micro	Exogenous	aggregators in HO
	Endogenous	robot model in RC
Substrate relative to Observer	Exogenous	Ext-Obs of a robot's model in RC
	Endogenous	robot's M_{Mod}-obs of its model in RC
Substrate Distribution	Centralised	collective opinion in CD
	Distributed	decision-makers' models in CD
Substrate Coupling	Tightly-coupled	individual models in CD and RC
	Loosely-coupled	collective models in CD
	Mobile	collective models of robots in RC
Encoding	Symbolic	models in HO, RC, CD
	Structural	robots' space distribution in RC resources' distribution in CD pheromone path in AF
Scope	Whole	top-models in HO, RC, CD
	Part	mid-models in HO, RC, CD
Observed Scope	Whole	models in HO
	Part	mid-models in RC, CD
Uniformity	Uniform / equivalent	substance concentration in HO ground-truth models in RC
	Heterogeneous	out-of-sync. oscillators, in HO partial robot models in RC

All four combinations of exogenous and endogenous macro-substrate are possible, relative to its micro-substrate and observer-substrate. E.g. in the HO case, macro-objects of mid-aggregators are exogenous relative to both top-aggregator observers and micro-oscillator objects. In CD, the agents' O_i models are exogenous to the observed resources. Conversely, in RC, robot models (macro-objects) are endogenous to the robots, which play the double roles of macro-observers and micro-objects. An external observer (Ext-obs) of the robots' models is exogenous to these models.

We also consider the macro-object's substrate continuity within its definition domain (e.g. spatial, temporal, numeric scope). We refer to this as the **substrate distribution**, which may be *centralised*, when the substrate's domain is continuous, or *distributed*, when it contains gaps. E.g. the top-average model in HO and the collective-model (O_{Col}) in CD are centralised; whereas the set of robot count-models in RC, or set of O_i models in CD are distributed.

Finally, we consider the **substrate coupling**, or flexibility with respect to its constituent parts. The coupling strength is a relational property that determines the extent to which replacing a part implies changes in related parts (as in software systems). In a *tightly-coupled* substrate, replacing a component necessitates changes in its interrelated components; whereas in a *loosely-coupled* substrate, components are less aware of each-other's changes. In *mobile* substrates, components continuously change locations, hence their interaction graph. E.g., all individual models in our case studies are tightly-coupled to their substrates. Still, an oscillator's concentration has a loosely-coupled, or mobile molecular substrate (assuming the substance is in liquid or gas state). Also, the entire set of models of all robots (RC) has a mobile substrate as robots move around. Other substrate dimensions and interpretations are possible, e.g. correlation degree among substrate parts [29].

C. Information encoding

In addition to considering the macro-object's substrate relative to its observer and micro-objects, we also consider the macro-object's information **encoding** onto its substrate [19]. The encoding can be *structural*, where information is represented by the actual substrate structure, and hence strongly dependent on a particular substrate arrangement – e.g. the concentration (macro-object) of molecules (micro-objects) in a substance (HO). Alternatively, the encoding may be *symbolic*, based on a conventional representation reproducible across various substrates – e.g. aggregator models in HO may be encoded onto a sensor's screen, stored in memory or on paper.

D. Information scopes

The macro-object's **scope** indicates whether it represents the *whole* macro-phenomena issued from the micro-objects; or only a *part* of it. In terms of our conceptual framework (IV), a macro-object's scope depends on whether it represents the whole collective-scope of micro-objects (μ_{Col}-obj) or only part of it (M_{Col}-obj). The macro-object represents the scope directly if it is a macro-collective object (M_{Col}-obj); or indirectly, if it is a model (M_{Mod}-obj, generated by a macro-collective observer, M_{Col}-obs). E.g. in HO, mid-aggregators only represent a partial scope of all oscillators' states; whereas the top-aggregator represents (indirectly) the whole scope.

If an observer only perceives part of a macro phenomena, we need another reference to determine what the corresponding whole is. We do this by indicating: i) another observer that perceives the whole phenomena; ii) the whole phenomena that the macro-observer aims to model for adaptation; or iii) the concatenation of overlapping partial scopes of similar observers, resulting in a larger scope overall.

The macro-observer perceives an **observed-scope**, which includes the subset of micro-objects that the observer samples from the macro-object's scope. This depends on the observer's sampling method, which may be: *complete* – leading to a *whole* observed-scope; or *scattered* (e.g. probabilistic or search-oriented) – leading to a *partial* observed-scope. Both kinds of observed scope are relative to the macro-object's scope. E.g. In RC, robots sample their environment partially as they move; and in CD, agents only probe their vicinity.

E. Uniformity relative to similar observers

We analyse a macro-object's **uniformity** with respect to multiple observations from similar observers – i.e. using an equivalent sampling method; and featuring the same observation scale and scope as the macro-object. In this sense, a macro-object may be *uniform* (or *equivalent*), meaning that observations from similar observers would be sufficiently alike to lead to the same behaviour. Otherwise, the macro-object is considered *heterogeneous*. E.g., the concentration (macro-object) of a substance at equilibrium appears uniform with respect to multiple observations within different substance sub-volumes (partial scope). In contrast, local agent observations (partial scope) of environmental resources (CD case) are heterogeneous, as they may lead to different individual opinions O_i about resource adaptations. Yet, simultaneous readings of the collective opinion O_{Col} results in equivalent adaptations. When a single observer of a macro-object is present, we can assess uniformity by imagining that more were available.

VI. Conceptual analysis of case studies

A. Analysis overview

In each case study, we analyse the macro-types of several macro-objects and their observers, to highlight how focusing on different macro-objects from different perspectives can change the corresponding macro-types. We proceed via the following steps, providing: 1) an overview of the main agent roles, interrelations and models, to help position each macro-type analysis of observer-object pairs; 2) a detailed macro-type analysis of each observer-object pair, with an additional table summarising the macro-type dimensions of all analysed pairs; and 3) a brief comparative review of all analysed macro-types, highlighting the key points that differentiate the various observer-object perspectives.

For the detailed analysis of each case study, we first focus on the reference micro-collective object (S_0). Then, for each inter-scale abstraction in the upward flow ($S_s \rightarrow S_{s+1}$, $s = 0..1$), we analyse: the macro-collective object obtained via scope-reduction (a); and the macro-model object obtained via info-merging (b). Finally, for the feedback flow, we analyse the macro-model object at the top-scale (S_2) perceived by a lower-scale observer (S_1 or S_0). This gives a common sequence of macro-type analysis for the following observer-object pairs:

(0) μ_{Col}-obj$_{S_0}$ $\leftarrow$ Ext-obs: a reference macro-collective object at S_0 comprising the entire micro-collective state μ_{Col}-obj$_{S_0}$, viewed by an external observer (Ext-obs);
(1) M_{Col}-obj$_{S_0}$ $\leftarrow$ M_{Col}-obs$_{S_1}$: a macro-collective object at S_0, obtained via (a) scope-reduction from μ_{Col}-obj$_{S_0}$, by a macro-collective observer at S_1; which then creates, via (b) info-merging, the macro-model object at S_1;
(2) M_{Mod}-obj$_{S_1}$ $\leftarrow$ M_{Mod}-obs$_{S_1}$: the macro-model object at S_1, perceived by a macro-model observer at S_1;
(3) M_{Col}-obj$_{S_1}$ $\leftarrow$ M_{Col}-obs$_{S_2}$: the set of all macro-model objects at S_1, perceived by a macro-collective observer at S_2, (M_{Col}-obs$_{S_2}$), which then creates the macro-model object at S_2;
(4) M_{Mod}-obj$_{S_2}$ $\leftarrow$ M_{Mod}-obs$_{S_2}$: macro-model object at S_2 viewed by a macro-model observer at S_2;
(5) M_{Mod}-obj$_{S_2}$ $\leftarrow$ M_{Mod}-obs$_{S_1/S_0}$: macro-model object at S_2, perceived by a lower-scale macro-collective observer – e.g. M_{Mod}-obs$_{S_0}$, when feedback goes directly from S_2 to S_0; and M_{Mod}-obj$_{S_1}$ when it goes indirectly through S_1. In this latter case, we have an extra case (6): macro-model object at S_1, perceived by a macro-collective observer at S_0 (e.g. in HO analysis).

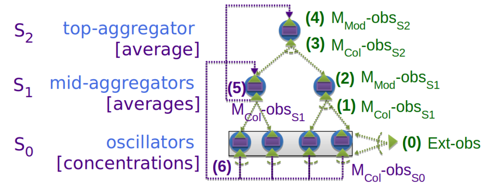

Fig. 5. Hierarchical Oscillators (HO) – analysis-cases II.(0)-(6)

In some case studies (e.g. HO), we also show how selecting a micro-collective object reference at a different scale (for analysis-case (0) above) changes the macro-type analysis.

B. Hierarchical Oscillators (HO)

1) Overview of agent roles and models: HO involves two agent types (Fig. 5): oscillators at S_0; and aggregators at S_1 and S_2 (mid- and top-aggregators, respectively). Each oscillator exposes its concentration to its mid-aggregator parent and changes its concentration based on feedback from this mid-aggregator. Aggregators hold average-concentration models (ac-models) based on the concentrations perceived from their children. Mid-aggregators also integrate into their ac-model$_{S_1}$ the feedback received from their top-aggregator, ac-model$_{S_2}$.

We show how the generic analysis-cases above characterise macro-types in the HO example: from oscillator concentrations (S_0 micro-reference) through averages (S_1, S_2, macro-objects) and back to concentrations (S_0).

2) Macro-types analysis (Table II):

I. **micro-object** (μ-obj): oscillator concentration; **micro-collective object** (μ_{Col}-obj): all oscillators;

I.(0) An external observer perceives the concentrations of oscillators at S_0. These latter form the analysed macro-object.

Macro-object: set of concentrations in all oscillators;

Macro-observer: external observer Ext-obs$_{S_0}$ of all oscillator concentrations, via complete sampling;

Macro-type: the macro-substrate is the set of substances of oscillators whose concentrations are observed. It is *endogenous* relative to the micro-substrate, which consists of the individual oscillator substances. It is *exogenous* relative to the external observer. The macro-substrate is also *distributed* and *loosely-coupled* across flexible oscillators. Information-encoding is *structural*, as concentration is a property of the substance substrate. The macro-object's scope is *whole*, as it includes all oscillator concentrations (in the μ_{Col}-obj). The observed-scope is also *whole*, via complete sampling; and so similar external observers perceive it in *equivalent* ways.

I.(1) A mid-aggregator observes the concentrations of two (out of four) oscillators – same as above, with reduced scope. It then averages these concentrations within an internal model.

Macro-object: concentrations of two oscillators;

Macro-observer: mid-aggregator, with complete sampling;

Macro-type: is the same as I.(0), except that it has a *part* scope (only two oscillators), which renders it *heterogeneous* to similar observers if oscillators are out-of-sync (assumed in Table II). Uniformity becomes $equivalent$ when oscillators synchronise (as observers perceive similar concentrations).

I.(2) A mid-aggregator observes its own model.

Macro-object: average concentrations in aggregator model;

Macro-observer: mid-aggregator observing its model;

Macro-type: is mostly opposite to I.(1) as the model (macro) is now *exogenous* to the oscillators (micro) and *endogenous* to its mid-aggregator (observer). The model is now *centralised* and *tightly-coupled* to its mid-aggregator. It is *symbolic* in its encoding, as it no longer depends on the substance substrate. Its scope remains *partial* relative to the two oscillators; and its observed-scope *whole* (via complete-sampling) – hence remaining *heterogeneous* for similar observers until oscillators synchronise, as in I.(1).

I.(3) The top-aggregator observes the two mid-aggregator models; and forms its own model (average).

Macro-object: mid-aggregators' two average models

Macro-observer: top-aggregator, complete sampling

Macro-type: is mostly the same as I.(2), except the macro-substrate of the two average models is now *distributed* and *loosely-coupled* (as mid-aggregators can be replaced); and the scope extends to *whole* (indirectly covering all four oscillators), giving *equivalent* uniformity.

I.(4) The top-aggregator observes its own model.

Macro-object: the top-aggregator's internal model

Macro-observer: top-aggregator, via complete sampling

Macro-type: is mostly the same as I.(2) – i.e., mid-aggregator observing its average model – except the scope becomes *whole* (all oscillators), giving *equivalent* uniformity.

I.(5) The mid-aggregator observes the top-aggregator's model (through the feedback information flow – top-down).

Macro-object: top-aggregator's model

Macro-observer: mid-aggregator, via complete sampling

Macro-type: is as in I.(4), except the mid-aggregator (observer) is now *exogenous* to the top-model (object).

I.(6) An oscillator observes its mid-aggregator's model (through the feedback information flow – top-down).

Macro-object: mid-aggregator's average model

Macro-observer: child oscillator, via complete sampling

Macro-type: is the same as I.(5), except uniformity becomes *heterogeneous* (as out-of-sync oscillators perceive different models from mid-aggregators, which merge their current average with the top-aggregator's average). Scope-wise, mid-models merge both the *whole* scope of the top-model with the *partial* scope of their child oscillators.

Next, we discuss three further analysis-cases (equivalent to (0)-(2) above, not shown in Fig. 5) to illustrate how selecting another reference micro-scale (S_0) may change macro-type analysis. Here, the reference micro-scale S_0 consists of molecule agents that lead to substance concentration in oscillators at S_1 (in turn used in analysis-case I.(0) above).

II. **micro-object** (μ-obj): molecule position in a substance; **micro-collective object** (μ_{Col}-obj): set of molecule positions in all oscillators;

II.(0) An external observer perceives the positions of all molecules in all oscillator substances.

Macro-object: molecule positions in all oscillators;

Macro-observer: external observer, via complete sampling;

Macro-type: the macro-substrate is the set of molecules of all substances – the same as the micro-substrate. As in I.(0), the macro-substrate is *endogenous* to the micro-substrate, *exogenous* to the external observer, *distributed* and *mobile* (assuming a liquid or gas substance). The macro-object's encoding is *structural*, as positions are properties of molecules. Both its scope and observed-scope are *whole*, being the same as the micro-collective object. Hence, similar observers perceive it in *equivalent* ways.

II.(1) Mid-aggregator observes all molecules in an oscillator

Macro-object: molecule positions in one oscillator;

Macro-observer: mid-aggregator perceiving all molecule positions in the oscillator, via complete sampling;

Macro-type: differs from that in II.(0) only by its *part* scope, leading to *heterogeneous* observations from similar observers (assuming these may perceive other oscillators).

II.(2) Mid-aggregator observes an oscillator's concentration

Macro-object: an oscillator's molecule concentration;

Macro-observer: mid-aggregator perceiving the oscillator's concentration, via complete sampling;

Macro-type: is mostly the same as that of II.(1)– the macro-substrate is still *distributed* and *mobile*, as it extends over multiple molecules. Yet, it is only *heterogeneous* if oscillators are out-of-sync (assumed in Table II); and $equivalent$ otherwise.

3) Notable analysis points: when observing substance concentrations (macro-objects) that are a property of substance molecules (collective micro-state), then the macro-substrate is always *endogenous* relative to the micro-substrate. When observing concentration models (macro-objects) their substrates are *exogenous* relative to substance molecules (micro-substrate). The observer's substrate is *exogenous* relative to the observed macro-substrate when aggregators observe other aggregators or oscillators; and *endogenous*, when aggregators observe their own models. Concentration models (in aggregators) have substrates that are *centralised*, *tightly-coupled* and *symbolic*; whereas actual concentrations have molecular substrates that are *distributed*, *mobile* (in liquids or gases) and *structurally* encoded. The macro-scope is *whole* when observing all oscillators, either at the lowest or the highest scales; and *partial* when only observing some of the oscillators, at the mid-scale. Within this scope, the observed-scope is always *whole*, as aggregators use a complete-sampling method. Uniformity is *equivalent* only when the scope is

I. μCollObj: substance concentration of all oscillators								
I.(0)	end	exo	dist	loose	str	whl	whl	eq
I.(1)	end	exo	dist	loose	str	prt	whl	ht
I.(2)	exo	endo	cntr	tight	sym	prt	whl	ht
I.(3)	exo	endo	cntr	tight	sym	whl	whl	eq
I.(4)	exo	endo	cntr	tight	sym	whl	whl	eq
I.(5)	exo	exo	cntr	tight	sym	whl	whl	eq
I.(6)	exo	exo	cntr	tight	sym	prt	whl	ht
II. μCollObj: molecule positions of all oscillators								
II.(0)	end	exo	dist	mob	str	whl	whl	eq
II.(1)	end	exo	dist	mob	str	prt	whl	ht
II.(2)	end	exo	dist	mob	str	prt	whl	ht

TABLE II
HIERARCHICAL OSCILLATORS (HO): MACRO-TYPE ANALYSIS

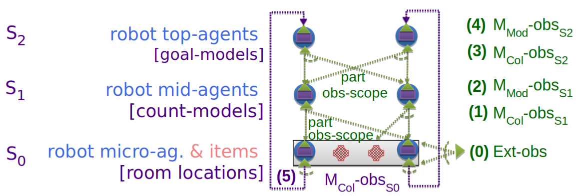

Fig. 6. Robot Collective (RC) – analysis series I.(0)-(5)

whole, including all oscillators; or when it is partial over synchronised oscillators that feature similar states.

C. Robotic Collective (RC)

1) Overview of agent roles and models: RC feature a single type of active entity: robots (Fig. 6). Items are passive and do not perceive information. Each robot hosts three agents: i) a *micro-agent* (S_0) that exposes its room location to observers – thus contributing to the lowest-scale micro-collective object; ii) a *mid-agent* (S_1) perceiving the robot's room location and the locations of other robots&items that it encounters (over a time-interval) forming a count-model; and, iii) a *top-agent* (S_2) perceiving its count-model and the encountered robots' count-models, forming a goal-model. This latter is seen by robot micro-agents (S_0) , which move accordingly.

2) Macro-types analysis (Table III):

I. **micro-object** (μ-obj): room location of robot or item; **micro-collective object** (μ_{Col}-obj): set of room locations of robots&items in all rooms;

I.(0) An external observer accurately perceives the locations of all robots&items in all rooms.

Macro-object: room locations of all robots&items;

Macro-observer: external observer of all room locations of robots&items, via complete sampling (e.g. video camera);

Macro-type: the macro-substrate is the set of all actual robots&items. It is the same as the micro-substrate and hence *endogenous* to it; while being *exogenous* to the external observer. It is *distributed* (across robots&items); and partly *mobile* (robots), partly *loosely-coupled* (replaceable items). The encoding is *structural*, as locations are properties of robots&items. Both the scope and observer-scope are *whole*, giving *equivalent* uniformity for similar external observers.

I.(1) A robot observes the locations of robots&items that it encounters while exploring the rooms.

Macro-object: room locations of all robots&items (I.0)

Macro-observer: a robot's mid-agent sampling robot&item locations as it moves across rooms; via scattered sampling.

Macro-type: is the same as in I.(0), except for its *partial* observed-scope (due to the robot's random encounters), making it *heterogeneous* to similar robot observers. We note that the macro-object's scope is still *whole*, as the robot could, in principle, encounter all robots&items.

I.(2) A robot observes its count-model, formed from perceived locations, cf. I.(1).

Macro-object: a robot's count-model (room distribution) of robots&items it encountered;

Macro-observer: a robot's mid-agent perceiving its count-model, via complete sampling;

Macro-type: the macro-substrate is the robot hosting the count-model. Hence, it is still *endogenous*, as in I.(1), relative to the micro-substrate, which consists of all robots in the collective. It becomes *endogenous* relative to the robot mid-agent observer, which holds the count-model. It also becomes *centralised* and *tightly-coupled* (within the robot); and uses *symbolic* encoding. The scope is partial (only modelling encountered robots&items), with *whole* observed-scope (entire count-model) and hence *heterogeneous* to different robot mid-agent observers.

I.(3) A robot observers the count-models of other robots, during their encounters.

Macro-object: the set of count-models of all robots

Macro-observer: a robot's top-agent sampling other robots' count-models as it encounters them; scattered sampling.

Macro-type: is mostly the same as I.(1) – as it covers the same *partial* observer-scope relative to the *whole* scope of the collective micro-state; except at the higher-scale of count-models rather than individual robot&item locations. It differs from I.(1) in that the encoding becomes *symbolic* and the substrate-coupling is fully *mobile* (only including robots).

I.(4) A robot observing its goal-model (abstraction flow).

Macro-object: a robot's goal-model;

Macro-observer: a robot's top-agent, perceiving its goal-model, via complete sampling;

Macro-type: is the same as I.(2) as it is a refined version of the robot's count-model, based on the others' count models.

I.(5) A robot observing its goal-model, via the top-down feedback flow; for deciding which room to move to next.

Macro-object: a robot's goal-model;

Macro-observer: a robot's micro-agent, perceiving the top-agent's goal-model; via complete sampling;

I. μCollObj: room locations of all robots&items								
I.(0)	end	exo	dist	mob & loose	str	whl	whl	eq
I.(1)	end	exo	dist	mob & loose	str	whl	prt	ht
I.(2)	end	end	cntr	tight	sym	prt	whl	ht
I.(3)	end	exo	dist	mob	sym	whl	prt	ht
I.(4)	end	end	cntr	tight	sym	prt	whl	ht
I.(5)	end	end	cntr	tight	sym	prt	whl	ht

TABLE III
ROBOT COLLECTIVE (RC) ANALYSIS

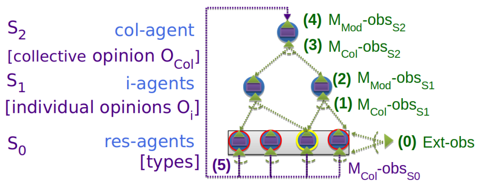

Fig. 7. Collective Decisions (CD) – analysis series I.(0)-(5)

Macro-type: is the same as I.(4) as it is the same macro-object (goal-model) perceived by an internal observer (micro-agent rather than top-agent); via complete sampling.

3) Notable analysis points: macro-substrates are *exogenous* when a robot sees the locations of other robots&items; and *endogenous* when the robot sees its own models. The macro-object's observation scope is always *whole*, as robots have, in principle, access to all other robots&items. Yet, the actual observed-scope is always *partial*, as robots only perceive a subset of robots&items via scattered-sampling. These scopes propagate through all scales, from observations of actual locations observations (micro-agents), to abstractions of these via count-models (mid-agents) and goal-models (top-agents). Finally, the encoding is *structural* when observing actual locations; and *symbolic* when observing models.

D. Collective Decisions (CD)

1) Overview of agent roles and models: CD features three kinds of agent (Fig. 7). Res-agents (S_0) are resources in the environment, exposing their types to observers (i-agents) and changing their types depending on the collective-opinion O_{col} (of the col-agent). Individual-agents (i-agents, S_1) observe local resource-agents and form an individual-opinion model (O_i-model). A collective-agent (col-agent, S_2) observes the set of O_i-models and forms a collective-opinion model (O_{col}-model), which is then seen by resource-agents.

2) Macro-types analysis (Table IV):

I. **micro-object** (μ-obj): types of res-agents;
micro-collective object (μ_{Col}-obj): set of res-agent types in the entire environment;

I.(0) An external observer perceives all typed resources (res-agents) in the environment.
Macro-object: all typed res-agents in the environment;
Macro-observer: external observer; complete sampling;
Macro-type: the macro-substrate onto which the resource types are encoded consists in the actual resources. It is the same as the micro-substrate and hence *endogenous* to it; while being *exogenous* to the external observer. The macro-substrate is *distributed* and *loosely-coupled* across replaceable resources (res-agents). Information-encoding is *structural*, as observed types are properties of resources. The scope is *whole* (including all res-agents), with *whole* observed-scope (external observer's complete sampling), giving *equivalent* views for similar external observers.

I.(1) An individual agent (i-agent) observes the types of resources in its neighbouring environment.
Macro-object: typed res-agents in an i-agent's vicinity;
Macro-observer: i-agent observing res-agent types in its vicinity; scattered sampling;
Macro-type: is almost the same as I.(0), as only the observer changes from external to i-agent. The difference is that both the scope and observed-scope become *partial* (only including the i-agent's vicinity, sampled randomly); hence leading to *heterogeneous* observations from similar i-agents.

I.(2) An individual agent (i-agent) observes its own model (O_i-model), formed from previous resource type observations.
Macro-object: O_i-model of dominant res-agent type;
Macro-observer: i-agent perceiving its O_i-model;
Macro-type: is mostly opposite to I.(1), except for its scope, which remains *partial*. The O_i-model is an *exogenous*, *centralised*, *tightly-coupled*, *symbolic* representation of the observed set of res-agent types in I.(1); and is *endogenous* to the i-agent observer, which sees it with *whole* observed-scope.

I.(3) The top-scale collective agent (col-agent) observes all individual models (O_i-model) of individual agents.
Macro-object: the set of O_i models of all i-agents;
Macro-observer: col-agent, with complete sampling;
Macro-type: the macro-substrate (i-agents) is *exogenous* relative to its micro-substrate (res-agents) and the observer (col-agent); *distributed* and *loosely-coupled* (over changeable i-agents). Information is encoded *symbolically*. The scope is *whole*, if assuming that overlapping O_i models cover the entire environment; and *partial* otherwise. As the observed-scope is *whole*, similar col-agents would perceive *equivalent* models if the scope were *whole*; and *heterogeneous* otherwise.

I.(4) The col-agent observes its O_{col}-model (abstract flow).
Macro-object: col-agent's O_{col}-model;
Macro-observer: col-agent; complete sampling;
Macro-type: is the same as I.(2) (i.e. i-agent viewing its O_i-model), as O_{col} is a higher-scale representation of the O_i set. Its scope takes the value of the O_i-models set I.(3): *whole* if i-agents cover the environment and *partial* otherwise; giving *equivalent* or *heterogeneous* perspectives, respectively.

I.(5) Resources (res-agents) perceive the actions issued by the col-agent, based on its O_{col}-model (feedback flow).

I. μCollObj: resource types in the entire environment								
I.(0)	end	exo	dist	loose	str	whl	whl	eq
I.(1)	end	exo	dist	loose	str	prt	prt	ht
I.(2)	exo	end	cntr	tight	sym	prt	whl	ht
I.(3)	exo	exo	dist	loose	sym	whl / prt	whl	eq / ht
I.(4)	exo	end	cntr	tight	sym	whl / prt	whl	eq / ht
I.(5)	exo	exo	cntr	tight	sym	whl / prt	whl	eq / ht

TABLE IV
COLLECTIVE DECISIONS (CD) ANALYSIS

Macro-object: col-agent's O_{col}-model, as in I.(4);

Macro-observer: res-agent; complete sampling;

Macro-type: is the same as I.(4), as only the observer changes (from col-agent to res-agent). Hence, the substrate becomes *exogenous* relative to the observer.

3) Notable analysis points: the macro-substrate is always *exogenous* when agents of different types observe each other (i.e. col-agents observe i-agents, which in turn observe res-agents, which finally observe the col-agent). Conversely, the substrate is always *endogenous* when an agent observes its own model (i.e. col-agent observing its O_{col}-model and i-agents their respective O_i-models). The substrate is *distributed* and *loosely-coupled* when an agent observes several other agents (e.g. col-agent observing i-agents, i-agents observing resources); and *centralised* and *tightly-coupled* when observing a single entity (e.g. agents observing their own model, or res-agents observing the col-agent model). The encoding is *structural* when observing resources types (encoded directly onto resources) and *symbolic* when observing models (substrate-agnostic). The scope is *partial* when i-agents can only observe their local environment. At higher scales (col-agent), it may become *whole* if the overlapping partial scopes of i-agents cover the entire environment. The observed-scope is always *whole* except when i-agents only observe the *partial* resources in their vicinity. Uniformity is equivalent only when both scope and observed scope are *whole*.

VII. DISCUSSION

A. Cross case-study analysis

Across each case study, we analysed the macro-types of several macro-objects, generated at increasing representation scales from micro-agents at the lowest scale (abstraction flow). We then analysed the macro-object at the top-scale observed by agents at lower scales (feedback flow). These examples concretely highlighted different types of macro-entities and how they could be integrated within an MSFS. E.g., macro-collective objects at S_0 were *structural*, *distributed* and *mobile* (or *loosely-coupled*); whereas their macro-models at S_2 were *symbolic*, *centralised* and *tightly-coupled*. Intermediate macro-models at S_1 covered *partial* scopes individually, and *whole* scopes collectively. Collective macro-models at S_1 were also *distributed* and *loosely-coupled*. Individual macro-models at S_1 featured either *whole* or *partial* observed-scopes.

While our case studies featured several common macro-type characteristics, these may differ in other MSFS. E.g., micro-collective objects (S_0) could also be *symbolic* and *distributed* (e.g. sensors); or *centralised* and *tightly-coupled* (e.g. large models) with macro-observers perceiving different parts. Also, macro-models could be *structural* and *loosely-coupled* – e.g. in HO, an oscillator's substance concentration (X) viewed as a model by the other substance (Y), which adapts to it (cf. causal emergence [8]). Or, macro-models could be *distributed* and only *partially* perceived by observers (e.g. in RC, a robot perceiving some of the other robots' models). Generally, scope-reduction engenders *heterogeneous* observations, unless all objects in the whole collective scope are *equivalent* (e.g. uniform substance concentration, or synchronised oscillators).

In exemplified cases, uniformity was a combined property of the macro-object's scope and observed-scope. If both scopes were *whole* then the macro-object was *equivalent*; otherwise it was *heterogeneous*. This would differ in the presence of observation errors or delays, which may lead to *heterogeneous* observations of objects with *whole* scopes. Also, *partial*-scope observers may use sampling-methods that give *equivalent* observations (e.g. picking the maximum value in a sub-set). When the observer's scope and sampling-method leads to *heterogeneous* models, this may engender different adaptations (e.g. in RC or CD, a 'perfect' external observer would propose different adaptations relative to robots or i-agents with partial views). In future work, we will also consider observers with different perception capabilities, meaning that they would view the same entity as different objects [21].

Several analysis cases fit two dimension categories, either: i) at the same time (e.g. in RC, the substrates of robots&items models are both *mobile* and *loosely-coupled*); or ii) depending on the system state when observed (e.g. HO observers view different concentration models as *heterogeneous* when oscillators are out-of-sync and *equivalent* when in-sync; in CD, the O_{coll} model covers the *whole* scope or only a *part*, depending on the combined coverage of the i-agents' *partial* scopes). Such cases may require more refined categorisation if differentiating them becomes important. When top-agents only cover a *partial* scope, micro-agents only adapt based on top-models that represent part of their collective state. More precise measures can be used to determine the extent of a scope's *partial* coverage relative to the *whole* scope.

Importantly, the conceptual macro-type framework did not explicitly include time aspects, which are essential in most MSFS (e.g. [17]). E.g., decreasing an observer's sampling frequency leads to scope-reduction; varying observation delays may increase model heterogeneity across observers; history-windows of various sizes may impact model accuracy (e.g. in case of random sampling, or for predictive models).

B. Assumptions and Practical Applicability

The proposed conceptual framework applies to any system implementing the MSFS pattern – i.e., where micro-entities coordinate their actions by sharing abstract information about their collective state. Specifically, this framework: helps to assess the macro-types of such collective state abstractions, applies to MSFS systems with heterogeneous macro-types, and can be extended or refined to handle the specificity of particular application domains – e.g. extending the binary scope options ('partial' or 'whole') to more gradual 'coverage' percentages; or expressing a dimension's dependence on the system's state or context. Assessing MSFS' abstraction quality, relative to its usage for adaptation, is out of this paper's scope and was the focus of a specific contribution [20].

In principle, categorising macro-types along the proposed dimensions will allow to characterise the resulting micro-macro feedback cycles in a qualitative manner – e.g. distributed, loosely-coupled versions may be more robust and adaptive, but slower, more imprecise and complicated than centralised ones; exogenous versions may encourage specialisation yet raise trust concerns; whole observation scopes with equivalent uniformity may provide more precise information yet at the cost of higher resource consumption and delays, while only providing benefits if paired with a correspondingly fast and accurate adaptation strategy. We started correlating macro-type dimensions with systemic qualities in previous work, e.g. [16], [17], [20]; to be completed by future research.

To select specific macro-type variants, system engineers should match such type-specific qualitative properties with the targeted system's constraints and requirements. Providing an explicit methodology for system design and evaluation based on such principles will be pursued in future work.

VIII. Conclusions and Future work

We proposed a conceptual framework to help discern various types of macro-entities in MSFS. First, we pinpointed key concepts relevant to micro- and macro-entities, and their relations (sec. IV) – i.e. agents; observer and object roles; collective information sources and models; and micro-macro abstraction processes based on scope-reduction and info-merging. Then, we relied on this conceptual basis to specify a multi-dimensional space for categorising macro-object types (sec. V) – based on their substrates, information encoding, scopes and uniformity; relative to their micro-information sources and/or to their macro-observers.

We illustrated the applicability of this conceptual macro-type framework via three MSFS case studies, showing its ability to distinguish between diverse macro-types. Exemplified cases covered a wide range of combinations of macro-type dimensions, without being exhaustive. We provided a high-level comparison of analysed case studies, and illustrated further cases not covered by these in the discussion (sec. VII).

In future work we will investigate less common cases to identify the conditions for their occurrence, or reasons for their impossibility. We will extend the framework to explicitly include time-related dimensions, e.g. observation frequency and delays. This proposal offers a conceptual basis for detecting and analysing micro-macro cycles in large CAS; and to help deal with the increasing complexity of modern environments.

References

[1] H. A. Simon, *The Architecture of Complexity*. Springer US, 1991.

[2] H. H. Pattee, *Hierarchy theory; the challenge of complex systems*. New York: G. Braziller, 1973.

[3] V. Ahl and T. F. Allen, *Hierarchy theory: a vision, vocabulary, and epistemology*. Columbia University Press, 1996.

[4] J. C. Flack, "Coarse-graining as a downward causation mechanism," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 375, no. 2109, p. 20160338, 2017.

[5] S. I. Walker, "Top-down causation and the rise of information in the emergence of life," *Information*, vol. 5, no. 3, pp. 424–439, 2014.

[6] W. R. Ashby, *The Ultrastable System*. Springer, 1960, pp. 80–99.

[7] A. Diaconescu, L. J. Di Felice, and P. Mellodge, "Multi-scale feedbacks for large-scale coordination in self-systems," in *IEEE 13th Intl. Conf. SASO*, 2019, pp. 137–142.

[8] F. E. Rosas et al., "Reconciling emergences: An information-theoretic approach to identify causal emergence in multivariate data," *PLOS Computational Biology*, vol. 16, no. 12, pp. 1–22, 12 2020.

[9] J. Kramer and J. Magee, "Self-managed systems: an architectural challenge," in *Future of Software Engineering*, 2007, pp. 259–268.

[10] D. Weyns et al., *On Patterns for Decentralized Control in Self-Adaptive Systems*. Springer Berlin Heidelberg, 2013, pp. 76–107.

[11] A. Schiendorfer, J.-P. Steghöfer, and W. Reif, "Synthesis and abstraction of constraint models for hierarchical resource allocation problems," in *Intl. Conf. ICAART*, 2014.

[12] J. Bourcier, A. Diaconescu, P. Lalanda, and J. A. McCann, "Autohome: An autonomic management framework for pervasive home applications," *ACM Trans. Auton. Adapt. Syst.*, vol. 6, no. 1, pp. 8:1–8:10, 2011.

[13] J. L. Fernandez-Marquez et al., "Description and composition of bio-inspired design patterns: a complete overview," *Nat. Comput.*, vol. 12, no. 1, pp. 43–67, 2013.

[14] M. Viroli et al., "From field-based coordination to aggregate computing," in *Coordination Models and Languages*. Springer, 2018, pp. 252–279.

[15] U. Schilcher, J. F. Schmidt, A. Vogell, and C. Bettstetter, "Swarmalators with stochastic coupling and memory," in *IEEE Int. Conf. ACSOS*, 2021, pp. 90–99.

[16] A. Diaconescu, L. J. Di Felice, and P. Mellodge, "Exogenous coordination in multi-scale systems: How information flows and timing affect system properties," *FGCS*, vol. 114, pp. 403–426, 2021.

[17] P. Mellodge, A. Diaconescu, and L. J. Di Felice, " Timing configurations affect the macro-properties of multi-scale feedback systems ," in *IEEE Intl.Conf. ACSOS*, online, Oct. 2021, pp. 100–109.

[18] K. L. Bellman and L. J. Goldberg, "Common origin of linguistic and movement abilities," *American J. of Physiology-Regulatory, Integrative and Comparative Physiology*, vol. 246, no. 6, pp. R915–R921, 1984.

[19] R. Feistel and W. Ebeling, "Entropy and the self-organization of information and value," *Entropy*, vol. 18, no. 5, p. 193, 2016.

[20] L.-J. Di Felice, A. Diaconescu, P. Zahadat, and P. Mellodge, "The value of information in multi-scale feedback systems," in *Frontiers in Complex Systems*, 2025, accepted (https://arxiv.org/abs/2505.11509).

[21] J. Von Uexküll, *A foray into the worlds of animals and humans: With a theory of meaning*. U of Minnesota Press, 2013, vol. 12.

[22] L. Floridi, "A defence of informational structural realism," *Synthese*, vol. 161, no. 2, pp. 219–253, 2008.

[23] L. Brillouin, "The negentropy principle of information," *Journal of Applied Physics*, vol. 24, no. 9, pp. 1152–1163, 1953.

[24] A. Diaconescu, L. J. D. Felice, and P. Mellodge, "An information-oriented view of multi-scale systems," in *IEEE ACSOS 2021, Companion Volume, SISSY Workshop*, 2021, pp. 154–159.

[25] A. Diaconescu, D. King, K. Bellman, C. Landauer, and P. Nelson, "Towards systems that dynamically change and evaluate abstractions," in *13th Conf. on Cognitive Situation Management (CogSIMA)*, 2023.

[26] E. Bonabeau and J. Dessalles, "Detection and emergence," *CoRR*, vol. abs/1108.4279, 2011. [Online]. Available: http://arxiv.org/abs/1108.4279

[27] A. Diaconescu, S. Tomforde, and C. Müller-Schloer, "Holonic cellular automata," in *ALIFE, Japan*. MIT Press, 2018, pp. 186–193.

[28] S. Frey, A. Diaconescu, D. Menga, and I. M. Demeure, "A generic holonic control architecture for heterogeneous multiscale and multiobjective smart microgrids," *ACM Trans. Auton. Adapt. Syst.*, vol. 10, no. 2, pp. 9:1–9:21, 2015.

[29] J. Stradner et al., "Algorithmic requirements for swarm intelligence in differently coupled collective systems," *Chaos, Solitons & Fractals*, vol. 50, pp. 100–114, 2013.