

A Cyber Sensor Model for Cyber-Physical-Social Systems

Olga Murdoch, University College Dublin, Ireland

Michael J. O'Grady, University College Dublin, Ireland

Gregory O'Hare, University College Dublin, Ireland

ABSTRACT

Engineering sustainable cyber-physical-social systems demand a transdisciplinary approach. Within an arbitrary domain, many systems, including those of the physical and cyber categories, may already be in-situ; however, heterogeneity permeates such systems, for example, differing protocols, data formats, among others. Heterogeneity is not a deliberate feature of an arbitrary system; rather, it is the cumulative result of pragmatic decisions that were made during design and is driven by many different factors, some of which may not be technological. Nonetheless, heterogeneity represents a critical obstacle for system designers as they seek to harness and integrate diverse system elements to deliver innovative services. This obstacle is acutely manifested in cyber-physical-social systems when collecting and fusing data for evidence-based decision-making; social and human-derived data exacerbate the problem. This paper proposes a programming model for fusing information sources in cyber-physical-social systems. The efficacy of the model is validated via a usability analysis.

KEYWORDS

Applied Software Engineering, Cyber-Physical-Social Computing, Human Factors, Programming Models

INTRODUCTION

Cyber-Physical Systems (CPSs) (Delicato et al., 2020; Liu et al., 2017) envisage a tight coupling between computing processes and the physical world. Engineering efficient, robust and secure CPSs pose many challenges; minor faults in design may result in catastrophic and tragic events in the physical world, as documented experiences with autonomous vehicles testify. Nonetheless, CPS are an indispensable and often invisible enabler of many services in the modern world. As the complexity of CPSs invariably increases, a need for more sophisticated models, methodologies and tools will be needed (Lee, 2015). For example, Intelligent CPSs (iCPSs) advocate embedding artificial intelligence techniques in the feedback loops for enabling decision-making in next-generation CPS (Vijayakumar et al., 2019). A complementary approach, that of Digital Twins (Rasheed et al., 2020), advocates constructing replicas of physical processes; in this way, more sophisticated models of reasoning and learning may emerge.

CPSs may leverage pre-existing cyberinfrastructures by incorporating them into their technology stack. The Sensor Web (Zhang et al., 2018) is an exemplar of such infrastructure; it enables programmatic access to networks of heterogeneous sensors for use in system development, sometimes in systems with global reach. Thus, when building CPSs, software engineers can call on an array of tried and trusted technologies such as Wireless Sensor Networks (WSNs), middleware, and Internet

DOI: 10.4018/IJAEIS.20210101.oa6

This article, published as an Open Access article on February 26th, 2021 in the gold Open Access journal, the International Journal of Agricultural and Environmental Information Systems (converted to gold Open Access January 1st, 2021), is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

of Things (IoTs) along with a range of mature standards, for example, those of the Open Geospatial Consortium (OGC). The net result is that the CPSs concept has been demonstrated in a wide range of domains including environmental monitoring (Mois et al., 2016), landslide detection (Liu et al., 2019), smart agriculture (Liu et al., 2020), health (Ashutosh Sharma et al., 2019) and even space technologies (Akyildiz & Kak, 2019).

Cyber-Physical-Social Systems (CPSSs) are, in contrast to their CPSs peers, less-well developed. Such systems offer an intriguing vision – the intrinsic incorporation of the human dimension in a variety of facets into system operation. Such incorporation may be relatively passive, for example, data or information provision, or may involve active human-in-the-loop participation that enables human-automation symbiosis as envisaged by Industry 4.0 (Rauch et al., 2020). Just as CPSs may leverage the Sensor Web, a CPSS may likewise leverage the Social Web (Russell & Klassen, 2018). In practice, challenges with data collection arise, especially if a big data solution is envisaged. However, even within relatively constrained circumstances, difficulties arising from a lack of software engineering toolkits and the inherent complexity. This paper demonstrates that in viewing CPSSs as an extension of CPSs, rather than a singular paradigm in its own right, core technologies that underpin CPSs may be refactored and augmented to form an intuitive foundation for either or both paradigms.

This paper is structured as follows. A review of the state-of-the-art is presented in the next section. A model for a CPSS incorporating social web precepts is then proposed. An implementation of this model is then validated using a cohort of software engineers. After discussing the results of the evaluation, the paper is concluded.

BACKGROUND

Cyber-Physical computing is considered as the successor to embedded systems and WSNs that traditionally underpin pervasive computing research. Definitions of cyber-physical computing primarily focus on the technical integration of cyber and physical systems, focusing on the communication capabilities of cyber systems and the benefits and complexities of such integration (Bordel et al., 2018; Y. Liu et al., 2017). The human element of cyber-physical computing underpins the concept of cyber-physical world convergence. Such convergence results in a paradigm shift in which “real-world components interact with cyberspace via sensing, computing and communication elements” causing information to flow “from the physical to the cyber-world and vice-versa, adapting the converged world to human behavior and social dynamics” (Conti et al., 2012, p. 2). Such a view is consistent with the definition of CPSSs offered by other researchers. Wang (2010) defines CPSs as “the tight conjoining of and coordination between computational (or cyber) and physical resources” p. 85, and further defines Cyber-Physical-Social systems as being cyber-physical systems “tightly conjoined, coordinated and integrated with human and social characteristics”. Sheth et al. (2013) define physical-cyber-social computing as a paradigm that “encompasses a holistic treatment of data, information, and knowledge from the physical-cyber-social “worlds to integrate, correlate, interpret, and provide contextually relevant abstractions to humans” p. 78. Physical-Cyber-Social computing is viewed as a next-generation paradigm which builds on ‘cyber-physical systems’, ‘socio-technical systems’, and ‘cyber-social systems’. Such a view explicitly acknowledges the diversity of technologies and infrastructures that must be seamlessly integrated if the vision of CPSSs is to become a reality.

CPSSs are at a nascent stage of their development lifecycle; although prototype CPSSs are documented (see, for example, Zhang et al., 2018), nonetheless, best practice in design and implementation is non-existent. One approach advocates the use of pre-existing methodologies from embedded systems and CPSs; one example of this approach has produced a system-level design framework (Zeng et al., 2020). Likewise, Zhou et al. (2020) have proposed a virtualization architecture, augmented with an integrated framework of caching, computing and networking. Following a theme of software reuse, Wang et al. (2020) advocate a Service Oriented Architecture (SOA) approach that allows the reuse of both business logic and software components, thereby enabling service composition

where pre-existing services can be dynamically selected, integrated and invoked in response to domain needs. A variation of SOA, microservice architectures, emphasize very focused, lightweight services; a choreography-driven microservice composition approach for CPSSs is proposed by Dai et al. (2020). Holistically, CPSSs are inherently trans-disciplinary and trans-domain; thus, the very nature of CPSSs may curtail the effectiveness of domain-specific processes. Simulation is a case in point; in response, Sánchez et al. (2017) have proposed a methodology for the design of co-simulation tools for CPSSs.

Conceptually and in practice, integrating the human dimension represents the singular most formable challenge facing CPSSs. Social media offers a human-centric but viable source of information concerning the real-world; such information can be of immense value to CPSSs, although the volume and diversity may pose significant problems. One response has been the design of a Country-Level Micro-Blog User (CLMB) behavior and activity model for use in CPSSs (Yang et al., 2019). Other research, the Social Sensor Cloud (Zhu et al., 2018), envisages an amalgamation of social networks and sensor networks modelled on the IoT but enabled via the Cloud.

Contribution

CPSSs have a particular resonance with environmental information systems. Conventional environmental sensing networks, either legacy or IoT-enabled, excel at capturing objective, time-series data at various spatial and temporal resolutions. However, the human-dimension is not captured; the social web offers a partial remedy in that the human experience, albeit subjective, can often be extracted from various social media data streams. Mining such streams enable the application of natural language processing, text analysis, and computational linguistics to extract meaning, through semantic or sentiment analyses, for example. In this way, concepts such as smart environments, smart farms, and smart homes may become more human-centric. Nonetheless, collecting, representing and fusing data to deliver human-centric and intelligent CPSSs is a formidable undertaking (Wang et al., 2019).

The approach described in this paper differs from other approaches to CPSSs development in that it advocates harnessing the middleware paradigm for managing both the physical and social dimension of an arbitrary CPSS. The business community has long applied middleware for the integration of different services; in the IoT and sensor communities, it is a proven method for enabling sensor abstractions and managing network heterogeneity (Martínez et al., 2017; Ngu et al., 2016). Usability will serve as the metric of choice for validating the approach.

MODELLING A CYBER-PHYSICAL-SOCIAL SYSTEM

At present, the “Cyber-Physical” dimension of CPSSs may be regarded as mature, especially when developments in IoTs are considered. This situation contrasts with the “Cyber-Social” dimension, which is less well developed despite the proliferation of Web2.0/3.0 and social media platforms. Remedying this deficiency is fundamental to delivering mature CPSSs thus, the conceptual design being proposed focuses on this element. The following principles underpin the design:

- The mature model of WSNs, fundamental to CPS, is complemented and replicated in so far as possible;
- Heterogeneity is intrinsic to CPSs and CPSSs; hence, the conventional strategy for managing diversity, that of middleware, is adopted;
- A uniform model of sensing that is tractable in both physical and cyberspace is a prerequisite; therefore, the core design entity is that of the Cyber Sensor (O’Grady et al., 2018).

In adhering to these principles, a holistic unified abstraction model emerges that delivers a pragmatic basis for enabling the rapid development of CPSSs prototypes and testbeds.

Data Sources for Cyber-Social Systems

Cyber-social systems constitute a rich array of disparate data sources; Figure 1 illustrates a taxonomy of these data sources; each is now further elucidated upon:

- **Static Resources:** These relate to sources where the information is, in theory, always valid, and does not change over time. Such sources are only queried in response to a need, for example, when seeking concept definitions.
- **Evolutionary Resources:** These may be regarded as the dynamic equivalent of static resources, providing information that evolves slowly; Wikipedia is an exemplar.
- **Event-driven Resources:** These resources constitute web sites, services or applications that consistently provide new information in response to real-world events. IoT and Sensor Web are exemplar.
- **User-driven Resources:** Such resources are comprising the Social Web, that is, social networking platforms, blogs amongst others.
- **Computational Resources:** These may be regarded as web services that provide data, given some contextual data as input.

Sensor Abstractions for Cyber-Physical-Social Systems

Cyber sensors constitute a logical abstraction for modelling the diverse data sources encountered in CPSSs and the monitoring of these sources in cyberspace. This abstraction embraces social, virtual and physical sensors. It may be broadly defined as a software component that monitors that particular environment which it is (programmatically) connected to, thus acquiring context of relevance to an arbitrary application. In terms of application development, sensor networks are usefully modelled in terms of node, network and infrastructure abstractions, as explained in the next section. Cyber Sensors are node-level abstractions. Extended this concept of abstraction result in the following typology:

- A Location Sensor acquires data from the perspective of a defined location. A location may be a geographical region or a personal space, for example, but it is always defined by a set of geographical coordinates.
- An Entity Sensor acquires data from the perspective of a defined entity. An entity can be a physical object or device in the real-world, or it may constitute a concept that represents a phenomenon, event or topic. An entity is described using keywords that capture its semantic meaning, for example, temperature.
- A User Sensor acquires data from the perspective of a defined user. A user in a Social Web context is a digital representation of a person, for example, a Facebook or Twitter profile.

Figure 1. Taxonomy of WWW-based sources of context

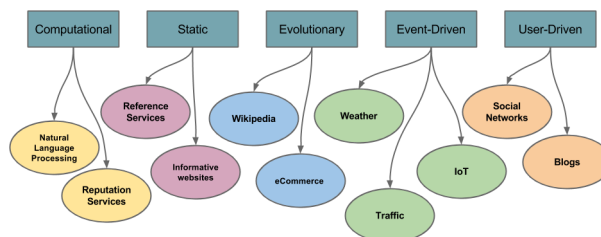


Table 1. Cyber Sensor Programming Abstractions

Vertical	Logical	Regional	Social
Node-Level	Entity Sensor	Location Sensor	User Sensor
Network-Level	Entity-Focussed Sensor Network	Location-Focussed Sensor Network	User-Focused Sensor
Infrastructure-Level	Entity-Focussed Sensor Infrastructure	Location-Focussed Sensor Infrastructure	User-Focussed Sensor Infrastructure

These categories of cyber sensors enable targeted monitoring of the various data sources necessary for CPSSs as outlined above.

Individual sensors generally have a 1-to-1 relationship with a data source. The value of a sensor network lies in its totality; in other words, the whole is greater than the sum of its parts. In practice, middleware is fundamental to realizing this value, enabling a process of discovery of both sensors and services. In this way, new services can be quickly composed, with added value being derived from the novelty of the composition. For software engineers, harnessing the pipeline design pattern offers one intuitive approach for chaining different services within middleware. However, a prerequisite for a Cyber Sensor Network is that of a programming model.

A Programming Abstraction for Cyber-Physical-Social Systems

When considering approaches to CPSs, well-established models used in the physical sensor network community offer a roadmap. For WSNs, high-level programming models may be categorized into two sub-levels: network-level and group-level abstractions (Sugihara & Gupta, 2008). Network-level abstractions are considered synonymous with macro-programming, in which an entire network is viewed and managed as a single abstract machine. Group-level abstractions are further classified into two categories of horizontal abstractions - regional and logical:

1. Regional abstractions define groups of sensor nodes based on the physical closeness of sensors. These neighborhood-based groups are typically defined by locality and consist of a node and its neighbors. While membership of nodes to a neighborhood-based group is mostly static, mobile and ad-hoc networks require dynamic management of such groups as nodes enter and leave the region.
2. Logical abstractions define groups of sensors nodes that are defined by the commonalities of logical properties between nodes, for example, node type and sensed modalities. Membership of nodes to logical groups is dynamic when dynamic property values determine membership.

Laukkarinen et al. (2012) define three levels of vertical abstraction - node, network and infrastructure; These programming abstractions are summarised in Table 1. Each is now considered.

Node Level Abstractions

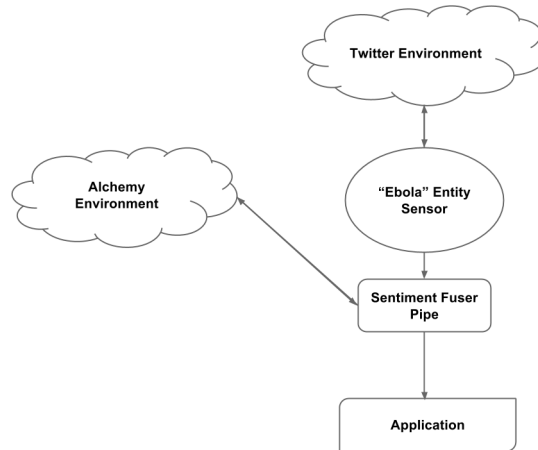
Node-level abstracts the resource-constrained, embedded, node hardware and communication protocols such that each node abstraction executes applications on each physical node. Node-level is the lowest level of abstraction, providing software engineers with a single point of access to data. Access is enabled through the provision of Entity and Location sensors for all Web resources, and User Sensors for all Social Web resources. Figure 2 illustrates this abstraction. In this example, the developer tasks the abstraction to monitor the sentiment of all “Keyword” entities found in the Twitter Environment. As a first step, an Entity Sensor for the Twitter environment is produced; this sensor is then tasked to produce information relating to “Keyword”. The next step in the processing chain is a call to the

IBM Alchemy API to detect sentiment; finally, the fused data is made available for consumption at the application layer.

Network-Level Abstractions

Network-level abstracts the distributed node network from data-interested users. At this level, distributed nodes cooperate for providing services such as data access through queries and data-processing in-network through aggregation and fusion. Network-level abstractions provide developers with a single point of access to data from entities, locations, or users identified as being relevant to an application at runtime. Figure 3 illustrates this abstraction, depicting a use case where a developer configured a network abstraction to monitor the sentiment of all Twitter users related to the entity “Keyword” (for example, ‘ebola’). In this example, an Entity Sensor is created to monitor “Keyword” entities on Twitter. This sensor is considered the ‘base sensor’ of the homogenous network. For each entity identified, a unique User ID is extracted and used in the generation of a User Sensor to monitor that user’s presence in the Twitter Environment. A call to the AlchemyAPI (via Sentiment Fuser Pipe) is then made to detect sentiment; a unique process (pipe) is created for each User Sensor generated.

Figure 2. The Node Level Abstraction for a Cyber Sensor data flow



Infrastructure-Level Abstractions

Infrastructure-level abstracts multiple heterogeneous sensor networks behind one interface. Such abstractions provide the same functionality as Network-Level abstractions but without tying the developer to a specific environment, for example, Twitter. Figure 4 illustrates this abstraction; here, the software developer wants to monitor the sentiment of Twitter users in countries with a hot climate. The base sensor in this example is an Entity Sensor monitoring the entity “hot” in a Weather environment. This Entity sensor produces information relating to all locations that are related to the entity “hot”. From this base sensor, location information is extracted and used in the generation of Location Sensors to monitor each detected location. From these (Twitter) Location Sensors, user ID information is extracted and used in the generation of User Sensors from which sentiment information is retrieved.

Implementation

For implementation, a logical strategy is to adopt a proven sensor middleware platform that is tried and trusted for the physical sensors and adheres to a Sensor Web paradigm. Such a platform can then

Figure 3. Network Level Abstraction for Cyber Sensor data flow

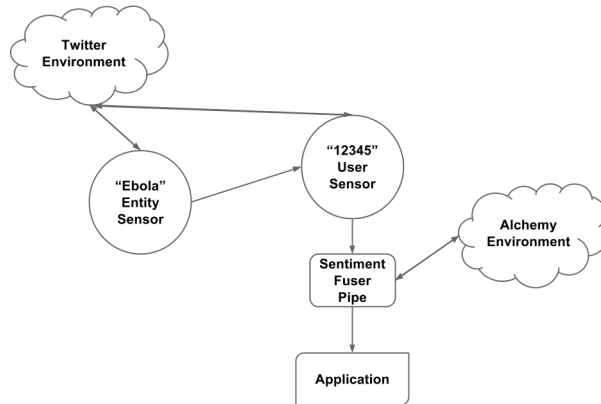
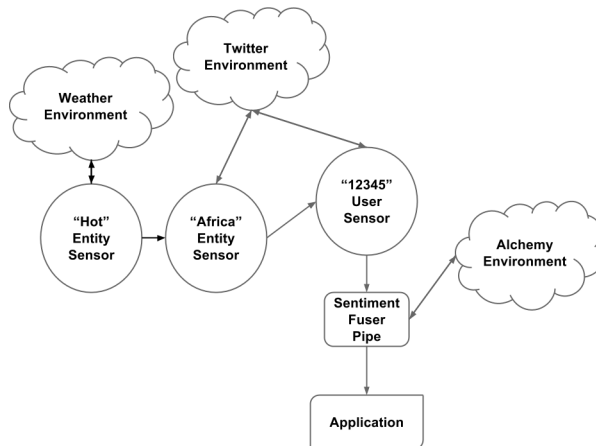


Figure 4. Infrastructure Level Abstraction for Cyber Sensor dataflow



be augmented with cyber-social-sensing functionality, as outlined in the previous sections. Thus, the design is implemented as an extension to a mature Sensor Web middleware, SIXTH (Carr et al., 2013). Within this extended implementation, cyber sensors are treated as equal citizens to their physical sensor counterparts. This equality facilitates the rapid development of CPSSs through a unified interface that shields developers from the underlying heterogeneity. Extensibility is facilitated by the definition of new node abstractions in the core middleware platform.

To illustrate the use of the API, consider a scenario where virus monitoring is needed. A proxy for environmental monitoring on the ground is that of social media; the translation is not exact, but it does serve as an indicator of trends. Thus, a Cyber Sensor is commissioned that monitors the “twitter” environment for the keyword “virus”. In this case, an Entity Sensor is required:

```
Sensor mySensor = createEntitySensor("twitter");
mySensor.task("virus");
```

However, if the resultant data is needed for decision making, information concerning its quality is essential. Assessing sentiment offers an indicator; the AlchemyAPI service may be harnessed

to aggregate sentiment information relating to “virus”. To interact with this Web service, a pipe is created; this pipe fuses sentiment information with the sensor (twitter) data stream in real-time:

```
Pipe sentimentPipe = CreateEntityAggregatorPipe('Alchemy ', 'sentiment');
```

However, a sensor stream must be specified for the pipe to act on, thus:

```
mySensor.applyPipe(myPipe);
```

Now that the Cyber Sensor is fully configured, it can be initiated:

```
mySensor.start();
```

Middleware usually directs its output to either a database or the application layer; output can also be directed to a console if needed. Figure 5 illustrates the high-level architecture of this middleware. Sensors and pipes are instantiated, configured and deployed according to the needs of the application domain in question. A task handler enables the definition of sensors and pipes, and their immediate configuration. On initialization, they become an intrinsic element of the service provision of the core middleware platform, using conventional services such as resource management and data storage.

EVALUATION: METHODS AND MATERIALS

The objective of this validation exercise is to assess the degree to which experienced software developers can intuitively grasp the cyber-sensor construct and harness a suite of customized middleware services to obtain data and undertake some processing on this data. A variety of metrics can be adopted to evaluate a CPS or CPSS. Quality, cost and security have been proposed for modeling iCPS deployment configurations (Gisselaire et al., 2019). Accuracy and timeliness are of particular importance in disaster management CPSs (Amit Sharma et al., 2019). For complex CPSs constituting many subsystems, an appropriate approach may be that of validating each system individually, as well as exploring the coupling between each of the subsystems (Legatiuk et al., 2017). Similarly, viewing a CPS through the lens of a system-of-systems approach offers a viable approach to analysis and design (Guariniello et al., 2020). For human-in-the-loop CPSSs, rapid prototyping offers a basis for analyses (Gil et al., 2020). Energy consumption, security-level, and user satisfaction may be considered vital dimensions when designing CPSSs (Zeng et al., 2017). For this discussion, the evaluation is user-centric, focusing on the usability of the middleware platform. In this way, the intuitiveness of the concepts, and the perceived applicability of the API design, may be assessed. Efforts may be directed at optimization after usability has been addressed.

Methodology

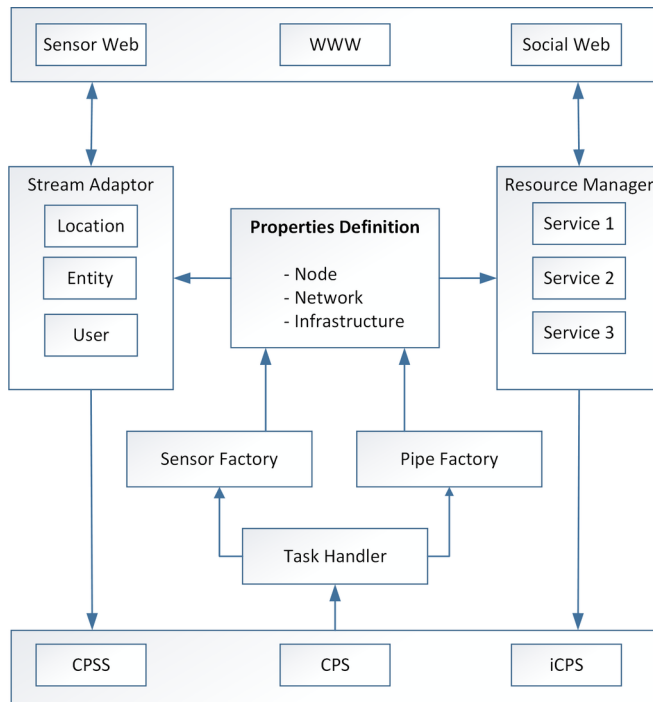
Four (4) discrete steps define the evaluation methodology.

Prior Experience

Prior experience of technology has the potential to influence how it may be perceived, experienced and utilized. Four categories of prior experience were identified:

1. Java programming experience;
2. Exposure and utilization of Web APIs;
3. Familiarity with sensor networking constructs and systems;

Figure 5. Model of Cyber Sensor enabled middleware for a CPSS



4. Utilization of middleware platforms.

Each category is assessed using a standard 5-point Likert scale ranging from no experience (0) to expert (5). For analysis purposes, subjects were categorized as being novices (scoring 0 to 3 on the Likert scale) or expert (scoring 4 to 5 on the Likert scale).

Understanding and Application

In order to evaluate each of the node, network and infrastructure abstractions, three distinct exercises were constructed. Each followed a similar pattern. A small tutorial (one page) was prepared that provided definitions for any particular abstractions that needed an explanation, e.g. “Entity Sensor”. Code snippets illustrated the creation and configuration of the corresponding classes and methods. Sample output was provided to illustrate what kind of results to expect. For each exercise, several discrete tasks were specified. On completion of each exercise, the participants complete a short questionnaire constituting 5-point Likert-scale questions.

Exercise 1: Sensors and Pipelines Abstractions

Tasks:

1. Create a ‘world weather’ location sensor to monitor Ireland.
2. Create a ‘twitter’ entity sensor to monitor ‘ebola’.
3. Create an ‘alchemy’ fuser pipe to monitor the sentiment of ‘ebola’ data.

Questions:

1. To what degree do you feel you completed the tasks?
2. To what degree do you feel you understand the Sensor concept?
3. To what degree do you feel you understand the Pipe concept?

4. To what degree do you feel you could apply the Sensor Pipeline concept to other problems?

Exercise 2: Sensor Network Abstractions

Tasks:

1. Create a 'twitter' network to produce user and location sensors related to the entity 'ebola'.
2. Create a 'world weather' network to produce location sensors for all locations related to the entity 'rain'.

Questions:

1. To what degree do you feel you completed the tasks?
2. To what degree do you feel you understand the Network concept?
3. To what degree do you feel you could apply the Network concept to other problems?

Exercise 3: Introduction to Sensor Infrastructures

Tasks:

1. Create an infrastructure to produce 'world weather' location sensors based on a 'twitter' entity sensor monitoring 'ebola'.
2. Create an infrastructure to monitor all users in countries where it is raining. You will need to monitor 'rain' from 'world weather' and produce 'twitter' location sensors accordingly. From this you can generate user sensors.

Questions:

1. To what degree do you feel you completed the tasks?
2. To what degree do you feel you understand the Infrastructure concept?
3. To what degree do you feel you could apply the Infrastructure concept to other problems?

Usability Assessment

For this evaluation, the classic view of usability as articulated by the International Organization for Standardization (ISO) 25010 product quality model is adopted; namely, the degree to which specified goals may be achieved with effectiveness, efficiency and satisfaction in a specified context of use. For assessing usability, the System Usability Scale (SUS) (Lewis, 2018) instrument was harnessed. SUS is a mature usability instrument and has been utilized in many domains. Most importantly, SUS produces reliable results from sample sizes as small as 8-12. SUS is not diagnostic; it is particularly useful for establishing baselines in agile development processes. Where usability issues recur, sophisticated techniques, for example, focus groups, might be needed.

Open Feedback

Provision of open feedback gives participants opportunities to raise those issues that may not have been captured in the formal evaluation process. Clarifications are offered, and persistent concerns recorded.

Evaluation Process

Twelve (n=12) researchers in computer science were requested to complete the three exercises. Each participant, by the nature of their research, was actively engaged in software development. Each session was limited to an hour; participants gave their time pro gratis. After verifying that the necessary software was installed and functioning on their laptops, they were given their instructions. Afterwards, they were debriefed by the lead author. Results were analyzed using SPSS. In the case of SUS, results were processed using Excel.

Table 2. Ranking of participants' expertise using a 5-point Likert scale, and their corresponding SUS score

Participant	Java	Web APIs	Sensor Networks	Middleware	SUS
1	4	1	0	4	97.5
2	5	0	0	0	92.5
3	5	5	5	5	72.5
4	4	4	3	5	70.0
5	4	2	4	3	52.5
6	4	4	4	4	75.0
7	5	1	5	4	85.0
8	5	2	2	4	67.5
9	4	3	1	3	90.0
10	4	0	0	0	90.0
11	5	3	5	5	82.5
12	5	4	0	4	87.5
Average	4.5	2.4	2.4	3.4	80.2

RESULTS

Initially, the reliability of the questionnaires was assessed; the Cronbach's alpha coefficient was calculated as 0.78. The SUS score resulting from the usability study is 80.2; as this is above the acknowledged threshold of 68, it is concluded that within the parameters of this exercise, the usability of the platform is above average.

Table 2 provides participant experience scores, using a 5-point Likert scale for knowledge of Java, APIs, sensor networks and middleware; individual SUS scores are provided for comparison. It is interesting to note that the highest scores are from participants with the least experience in Web APIs and Sensor Networks. Four of the participants with experience in sensor networks noted that the network and infrastructure-level abstractions made them feel as if they were 'losing control' over what sensors were being generated in the background. As middleware by its nature seeks to hide the intricacies of software, such observations may be expected. Likewise, these observations support the provision of higher-level abstractions for those with less experience. During the debriefing session, these participants were talked through how the more complex challenges may be solved using the node-level abstraction alone. Mainly, the participants were satisfied with the model as a solution for their level of expertise. However, three participants noted that they would like to apply the higher-level abstractions whilst having access to the 'raw' rules, rather than just applying the abstracted methods in the API.

In Table 3, participants rated their ability to complete the assigned task, understand the underlying concepts, and rate their ability to apply the concept in a real-life scenario. Again, the corresponding SUS scores are provided. Overall, the abstractions were understood, and the tasks mostly completed. The infrastructure abstraction seems to provide the most difficulties. Two scores fell below the 68 threshold at 52.5 and 67.5 respectively. The latter participant completed all tasks successfully but noted that better, and preferably graphical, documentation would make the concepts easier to understand more quickly. The participant also questioned the relationship between the concepts, wondering if the same problem could be solved using any of the three techniques. During the debriefing session, this was confirmed, and the participant stated that documentation to that effect would make the concepts

easier to grasp. The former participant was able to complete each task but struggled to select the appropriate invariant for adaptor names and variable names according to the reference sheet.

Two other participants suggested that it might be better to use static variables rather than ‘free text’. However, this is a feature of SOA architectures and is essential for maintaining the loose coupling between services thus ensuring the system remains extensible while maximizing the reusability of individual components. It is also a feature that developers using Web Services are accustomed. The participants that provided this feedback each gave themselves 3 out of 5, or less, for Web API experience.

All subjects professed significant experience in Java. Thus, Java could not be considered as a potential determinant of the user experience. However, familiarity with sensor constructs influenced perceptions. Those classifying themselves as novices in sensor technologies perceived the functionality as being well integrated - $U = 5.00$, $p < 0.05$. This positive perception suggests that such subjects found the notion of a sensor intuitive and found it easy to harness the APIs to fulfil the requested tasks. Those with sensor experience were more circumspect, perhaps expecting additional features including some that they may have had exposure to in the past. Novice users of middleware found the notion of a pipe somewhat difficult to comprehend; those with prior experience in working with middleware platforms were more confident they could apply the pipe concept to other problems - $U = 3.00$, $p < 0.05$.

During free feedback, a variety of issues were raised for clarification. However, the key recommendation that emerged was the inclusion of additional categories of pipes.

CONCLUSION

Harnessing a distributed sensing paradigm for modelling data sources for CPSSs is tractable, offering a novel model for effectively managing data complexity and diversity. Middleware is standard in both business information systems and sensor network development; thus, it offers a pragmatic foundation for CPSSs. However, a singular middleware approach may not suffice for CPSSs in all circumstances. Integrating core middleware precepts of fault-tolerance and load-balancing with a flexible microservice architecture encompassing cyber-sensor principles may allow for more scalability as well as enabling polyglot software development. In this way, the diversity and complexity of CPSSs may be addressed more sustainably going forward.

Table 3. Participants understanding of Cyber Sensor concepts, and their corresponding SUS score

#	Pipelines				Network			Infrastructure			SUS
	Understood				Understood			Understood			
	Complete	Sensors	Pipes	Application	Complete	Understood	Application	Complete	Understood	Application	
1	5	5	4	5	5	5	4	4	4	4	97.5
2	5	5	5	4	5	5	5	3	5	3	92.5
3	5	5	4	4	5	5	4	5	5	4	72.5
4	5	5	5	5	5	4	4	5	4	3	70
5	4	3	3	3	3	3	2	4	3	4	52.5
6	4	5	4	4	3	4	4	5	5	5	75
7	4	5	4	5	5	4	5	5	3	4	85
8	5	4	4	5	5	4	4	4	4	4	67.5
9	5	5	4	4	5	5	5	5	4	4	90
10	5	5	4	5	5	4	4	4	4	4	90
11	5	5	5	5	5	5	4	5	4	4	82.5
12	5	5	3	4	5	4	4	5	5	4	87.5

REFERENCES

- Akyildiz, I. F., & Kak, A. (2019). The Internet of Space Things/CubeSats: A ubiquitous cyber-physical system for the connected world. *Computer Networks*, 150, 134–149. doi:10.1016/j.comnet.2018.12.017
- Bordel, B., Alcarria, R., de Rivera, D. S., & Robles, T. (2018). Process execution in Cyber-Physical Systems using Cloud and Cyber-Physical Internet services. *The Journal of Supercomputing*, 74(8), 4127–4169. doi:10.1007/s11227-018-2416-4
- Carr, D., O'Grady, M. J., O'Hare, G. M. P., & Collier, R. (2013). SIXTH: A Middleware for Supporting Ubiquitous Sensing in Personal Health Monitoring. In B. Godara & K. S. Nikita (Eds.), *Wireless mobile communication and healthcare (Mobihealth 2012), Paris, France, November 21-23, 2012, revised selected papers* (1st ed., Vol. 61, pp. 421–428). Springer-Verlag Berlin Heidelberg. doi:10.1007/978-3-642-37893-5_47
- Conti, M., Das, S. K., Bisdikian, C., Kumar, M., Ni, L. M., Passarella, A., Roussos, G., Tröster, G., Tsudik, G., & Zambonelli, F. (2012). Looking ahead in pervasive computing: Challenges and opportunities in the era of cyber-physical convergence. *Pervasive and Mobile Computing*, 8(1), 2–21. doi:10.1016/j.pmcj.2011.10.001
- Dai, F., Mo, Q., Qiang, Z., Huang, B., Kou, W., & Yang, H. (2020). A Choreography Analysis Approach for Microservice Composition in Cyber-Physical-Social Systems. *IEEE Access: Practical Innovations, Open Solutions*, 8, 53215–53222. doi:10.1109/ACCESS.2020.2980891
- Delicato, F. C., Al-Anbuky, A., & Wang, K. I.-K. (2020). Editorial: Smart Cyber-Physical Systems: Toward Pervasive Intelligence systems. *Future Generation Computer Systems*, 107, 1134–1139. doi:10.1016/j.future.2019.06.031
- Gil, M., Albert, M., Fons, J., & Pelechano, V. (2020). Engineering human-in-the-loop interactions in cyber-physical systems. *Information and Software Technology*, 126, 106349. doi:10.1016/j.infsof.2020.106349
- Gisselaire, L., Cario, F., Guerre-berthelot, Q., Zigmann, B., Du Bousquet, L., & Nakamura, M. (2019). Toward Evaluation of Deployment Architecture of ML-Based Cyber-Physical Systems. In *2019 34th IEEE/ACM International Conference on Automated Software Engineering Workshops: 10-15 November 2019, San Diego, California* (pp. 90–93). doi:10.1109/ASEW.2019.00036
- Guariniello, C., Raz, A. K., Fang, Z., & DeLaurentis, D. (2020). System-of-systems tools and techniques for the analysis of cyber-physical systems. *Systems Engineering*, 23(4), 480–491. Advance online publication. doi:10.1002/sys.21539
- Laukkarinen, T., Suhonen, J., & Hännikäinen, M. (2012). A Survey of Wireless Sensor Network Abstraction for Application Development. *International Journal of Distributed Sensor Networks*, 8(12), 740268. doi:10.1155/2012/740268
- Lee, E. A. (2015). The past, present and future of cyber-physical systems: A focus on models. *Sensors (Basel)*, 15(3), 4837–4869. doi:10.3390/s150304837 PMID:25730486
- Legatiuk, D., Dragos, K., & Smarsly, K. (2017). Modeling and evaluation of cyber-physical systems in civil engineering. *Proceedings in Applied Mathematics and Mechanics*, 17(1), 807–808. doi:10.1002/pamm.201710371
- Lewis, J. R. (2018). The System Usability Scale: Past, Present, and Future. *International Journal of Human-Computer Interaction*, 34(7), 577–590. doi:10.1080/10447318.2018.1455307
- Liu, R., Zhang, Y., Ge, Y., Hu, W., & Sha, B. (2020). Precision Regulation Model of Water and Fertilizer for Alfalfa Based on Agriculture Cyber-Physical System. *IEEE Access: Practical Innovations, Open Solutions*, 8, 38501–38516. doi:10.1109/ACCESS.2020.2975672
- Liu, Y., Peng, Y., Wang, B., Yao, S., & Liu, Z. (2017). Review on cyber-physical systems. *IEEE/CAA Journal of Automatica Sinica*, 4(1), 27–40. <ALIGNMENT.qj></ALIGNMENT>10.1109/JAS.2017.7510349
- Liu, Z., Tsuda, T., Watanabe, H., Ryuo, S., & Iwasawa, N. (2019). Data Driven Cyber-Physical System for Landslide Detection. *Mobile Networks and Applications*, 24(3), 991–1002. doi:10.1007/s11036-018-1031-1
- Martínez, E., Toma, D. M., Jirka, S., & Del Río, J. (2017). Middleware for Plug and Play Integration of Heterogeneous Sensor Resources into the Sensor Web. *Sensors (Basel)*, 17(12), 2923. Advance online publication. doi:10.3390/s17122923 PMID:29244732

- Mois, G., Sanislav, T., & Folea, S. C. (2016). A Cyber-Physical System for Environmental Monitoring. *IEEE Transactions on Instrumentation and Measurement*, 65(6), 1463–1471. doi:10.1109/TIM.2016.2526669
- Ngu, A. H. H., Gutierrez, M., Metsis, V., Nepal, S., & Sheng, M. Z. (2016). IoT Middleware: A Survey on Issues and Enabling technologies. *IEEE Internet of Things Journal*, 1, 1. Advance online publication. doi:10.1109/JIOT.2016.2615180
- O’Grady, M. J., Murdoch, O., Carr, D., Collier, R., & O’Hare, G. M. P. (2018). The Cyber Sensor Network. In *2018 IEEE 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech* (pp. 386–389). IEEE. doi:10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.00076
- Rasheed, A., San, O., & Kvamsdal, T. (2020). Digital Twin: Values, Challenges and Enablers From a Modeling Perspective. *IEEE Access: Practical Innovations, Open Solutions*, 8, 21980–22012. doi:10.1109/ACCESS.2020.2970143
- Rauch, E., Linder, C., & Dallasega, P. (2020). Anthropocentric perspective of production before and within Industry 4.0. *Computers & Industrial Engineering*, 139, 105644. doi:10.1016/j.cie.2019.01.018
- Russell, M. A., & Klassen, M. (2018). *Mining the social web* (3rd ed.). O’Reilly Media.
- Sánchez, B. B., Alcarria, R., Sánchez-Picot, Á., & Sánchez-de-Rivera, D. (2017). A Methodology for the Design of Application-Specific Cyber-Physical Social Sensing Co-Simulators. *Sensors (Basel)*, 17(10), 2177. Advance online publication. doi:10.3390/s17102177 PMID:28937610
- Sharma, A., Rathee, G., Kumar, R., Saini, H., Varadarajan, V., Nam, Y., & Chilamkurti, N. (2019). A Secure, Energy- and SLA-Efficient (SESE) E-Healthcare Framework for Quickest Data Transmission Using Cyber-Physical System. *Sensors (Basel)*, 19(9), 2119. Advance online publication. doi:10.3390/s19092119 PMID:31067811
- Sharma, A., Singh, P. K., Sharma, A., & Kumar, R. (2019). An efficient architecture for the accurate detection and monitoring of an event through the sky. *Computer Communications*, 148, 115–128. doi:10.1016/j.comcom.2019.09.009
- Sheth, A., Anantharam, P., & Henson, C. (2013). Physical-Cyber-Social Computing: An Early 21st Century Approach. *IEEE Intelligent Systems*, 28(1), 78–82. doi:10.1109/MIS.2013.20
- Sugihara, R., & Gupta, R. K. (2008). Programming models for sensor networks. *ACM Transactions on Sensor Networks*, 4(2), 1–29. doi:10.1145/1340771.1340774
- Vijayakumar, V., Subramaniaswamy, V., Abawajy, J., & Yang, L. (2019). Intelligent, smart and scalable cyber-physical systems. *Journal of Intelligent & Fuzzy Systems*, 36(5), 3935–3943. doi:10.3233/JIFS-179108
- Wang, F.-Y. (2010). The Emergence of Intelligent Enterprises: From CPS to CPSS. *IEEE Intelligent Systems*, 25(4), 85–88. doi:10.1109/MIS.2010.104
- Wang, P., Yang, L. T., Li, J., Chen, J., & Hu, S. (2019). Data fusion in cyber-physical-social systems: State-of-the-art and perspectives. *Information Fusion*, 51, 42–57. doi:10.1016/j.inffus.2018.11.002
- Wang, S., Zhou, A., Yang, M., Sun, L., Hsu, C.-H., & Yang, F. (2020). Service Composition in Cyber-Physical-Social Systems. *IEEE Transactions on Emerging Topics in Computing*, 8(1), 82–91. doi:10.1109/TETC.2017.2675479
- Yang, P., Liu, J., Qi, J., Yang, Y., Wang, X., & Lv, Z. (2019). Comparison and Modelling of Country-level Microblog User and Activity in Cyber-physical-social Systems Using Weibo and Twitter Data. *ACM Transactions on Intelligent Systems and Technology*, 10(6), 1–24. doi:10.1145/3339474
- Zeng, J., Yang, L. T., Lin, M., Ning, H., & Ma, J. (2020). A survey: Cyber-physical-social systems and their system-level design methodology. *Future Generation Computer Systems*, 105, 1028–1042. doi:10.1016/j.future.2016.06.034
- Zeng, J., Yang, L. T., Lin, M., Shao, Z., & Zhu, D. (2017). System-Level Design Optimization for Security-Critical Cyber-Physical-Social Systems. *ACM Transactions on Embedded Computing Systems*, 16(2), 1–21. doi:10.1145/2925991

Zhang, J. J., Wang, F.-Y., Wang, X., Xiong, G., Zhu, F., Lv, Y., Hou, J., Han, S., Yuan, Y., Lu, Q., & Lee, Y. (2018). Cyber-Physical-Social Systems: The State of the Art and Perspectives. *IEEE Transactions on Computational Social Systems*, 5(3), 829–840. doi:10.1109/TCSS.2018.2861224

Zhang, X., Chen, N., Chen, Z., Wu, L., Li, X., Zhang, L., Di, L., Gong, J., & Li, D. (2018). Geospatial sensor web: A cyber-physical infrastructure for geoscience research and application. *Earth-Science Reviews*, 185, 684–703. doi:10.1016/j.earscirev.2018.07.006

Zhou, Y., Yu, F. R., Chen, J., & Kuo, Y. (2020). Cyber-Physical-Social Systems: A State-of-the-Art Survey, Challenges and Opportunities. *IEEE Communications Surveys and Tutorials*, 22(1), 389–425. doi:10.1109/COMST.2019.2959013

Zhu, C., Leung, V. C. M., Rodrigues, J. J. P. C., Shu, L., Wang, L., & Zhou, H. (2018). Social Sensor Cloud: Framework, Greenness, Issues, and Outlook. *IEEE Network*, 32(5), 100–105. doi:10.1109/MNET.2018.1800029

Olga Murdoch is a computer scientist at the Lean Six Sigma Green Belt, Agile and at Prince2 she is certified project manager. She is currently working in University College Dublin.

Michael O'Grady is a Senior Researcher at the School of Computer Science & Informatics at University College Dublin. His research interests include Ambient Intelligence, Mobile Computing and Multimedia Systems. He has spent over a decade working in the software and telecommunications industries. In his research career, he has authored and contributed to over 150 publications in the broad area of ambient Intelligence and has served in a variety of roles in conference organization and review committees. Dr O'Grady is a senior member of the ACM and IEEE.

Gregory O'Hare completed his studies at the University of Ulster graduating with a B.Sc, M.Sc and Ph.D. He held the position of Head of the Department of Computer Science at University College Dublin (UCD) 2001-2004. Prior to joining UCD he has been on the Faculty of the University of Central Lancashire (1984-86) and the University of Manchester (1986-1996). He is an Associate Professor within the School of Computer Science & Informatics at UCD. He has published over 400 refereed publications in Journals and International Conferences, 7 books and has won significant grant income (ca €28.00M). O'Hare is an established researcher of international repute. His research interests are in the areas of Distributed Artificial Intelligence and Multi-Agent Systems (MAS), and Mobile & Ubiquitous Computing, Autonomic Systems and Wireless Sensor Networks. He has supervised some 18 Ph.D and 29 M.Sc.s to completion in his career to date. He referees extensively for Journals, International Conferences and funding agencies including the European Commission, Enterprise Ireland, Science Foundation Ireland, the Netherlands Organisation for Scientific Research, the National Science Foundation (US). He serves on the Editorial Review Board of the International Journal of Agent Technologies and Systems (IJATS), International Journal of Mobile Human Computer Interaction (IJMHCI) and the International Journal on Knowledge Based Intelligent Engineering Systems. In 2003 he received the prestigious Cooperative Information Agents (CIA), System Innovation Award for ACCESS: An Agent Architecture for Ubiquitous Service Delivery. O'Hare is a Fellow of the British Computer Society, a Fellow of the Irish Computer Society a member of the ACM, AAAI and a Chartered Engineer. He is the Chair of the European Research Consortium on Informatics and Mathematics (ERCIM) Working Group on Sensor Web. He has also held a prestigious Science Foundation Ireland (SFI) Principal Investigator Award 2003-2007. He is one of the Principal Investigators and founders of the Science Foundation Ireland funded (€16.4M) Centre for Science and Engineering Technologies (CSET) entitled CLARITY: The Centre for Sensor Web Technologies (2008-2013). In 2008-2009 he secured a Visiting Research Fellowship to the University of Oxford. In 2010 he was awarded a Fulbright Scholar visiting position at the Computer Science and Artificial Intelligence Laboratory (CSAIL) at Massachusetts Institute of Technology (MIT).