STORY

added value of STORage in distribution sYstems

Deliverable 3.8
Updated report on interoperability guidelines

Revision 1
Preparation date .. 30-04-2019 (M48)
Due date 30-04-2019 (M48)
Lead contractor .... UCL
Dissemination level PU

Authors:
Paul Valckenaers – UCL

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646426

Project STORY - H2020-LCE-2014-3

STORY

Table of contents

1 IN-DEPTH INTEROPERABILITY – LESSONS LEARNED (EXECUTIVE SUMMARY) 3

2 INTRODUCTION 5

3 LESSONS LEARNED – SMART ENERGY CHALLENGES 6

4 EMBODIED AND SITUATED DIGITAL TWINS IMPLEMENTING THE ARTI ARCHITECTURE 8

4.1 THE ARTI REFERENCE ARCHITECTURE – THREE COLOURS 10

4.2 THE ARTI REFERENCE ARCHITECTURE – FOUR TYPES OF DIGITAL TWINS 12

4.3 THE ARTI REFERENCE ARCHITECTURE – SIMON'S AGGREGATES 17

4.4 SELF-MODELLING TWINS AND TWINS OF MENTAL STATES 20

5 SAMPLE SCENARIOS – ADDED VALUE FROM IN-DEPTH INTEROPERABILITY 21

6 SCIENTIFIC LAWS OF THE ARTIFICIAL: THE SCIENCE UNDERPINNING THE SOLUTION. 24

6.1 SIMON'S LAW : TIME-VARYING AGGREGATION HIERARCHIES 26

6.2 COMPLEX-ADAPTIVE SYSTEMS THEORY: AUTOCATALYTIC SETS OR CRITICAL USER MASS 27

6.3 DESIGN FOR THE UNEXPECTED: COMPOSE-ABILITY AS A MANDATORY REQUIREMENT 29

6.4 COLLECTIVE IMAGINATION: SELF-MODELLING IN THE SKILL SELECTION 30

6.5 REMARKS 31

7 CONCLUSIONS 31

8 ACRONYMS AND TERMS 32

9 REFERENCES 32

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646426

Project STORY - H2020-LCE-2014-3

1 In-depth Interoperability – lessons learned (executive summary)

This document addresses in-depth interoperability, translating and applying scientific insights going beyond the state-of-the-art [1]. It contains insights and guidelines on how to design and build in-depth interoperable components and subsystems that can be integrated with little effort and time. The guidelines allow the integration of components and subsystems – through commonplace interoperation – into an overall system while ensuring good performances. It will be accompanied by D3.7, which contains software that exemplifies (in source code) the material in this document.

Although this document is intended to be readable on its own, it is a follow-up on D3.6, which contained knowhow and information that existed prior to STORY. This document accounts for the lessons learned during the project. It also aims to focus on new information and avoid repeating what I know already in the domain of smart energy and industrial automation.

The lessons learned (and others remarking that there is little new under the smart energy sun as far as ICT is concerned) induce a position statement. ICT in smart energy, even in industrial automation, is locked in into legacy technologies. Technically, an ICT platform disruption is long overdue, but in practice, legacy ICT is deeply entrenched.

However, the new challenges in the energy domain, and their urgency, increase the pressure against the walls around the comfort zones of this legacy continuously. The cost for society of staying with legacy technology is growing. Moreover, discussing what can be done within this comfort zone of legacy technology will not bring anything new. Therefore, this document aims to contribute beyond those walls. In particular, when the disruption kicks in, it attempts to maximise the benefits that emerge from the disruption; we should not miss the opportunities that accompany a good crisis.

The key concept, discussed in this document, is the *blue collar digital twin*. Here. *blue collar* stresses the online role of the digital twin. In addition, *twin* informs us that this digital entity will not invent anything that does not exist in reality: a digital twin *only mirrors* its physical twin. Moreover, the document introduces four twin types: (1) Activity Types, (2) Resource Types, (3) Activity Instances, and (4) Resource Instances. This document justifies their existence (why we need those four types) in combination with dynamic aggregation.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646426

Project STORY - H2020-LCE-2014-3

2 Introduction

This document addresses *in-depth interoperability,* which differs from and is complementary to the more commonplace understandings of interoperability. Effective interoperation of systems requires (at least) three features: (1) syntactic interoperation, (2) semantic interoperation and (3) real-world or in-depth interoperation. Syntactic interoperation has systems exchange data and raw information. An example is "JSON over IP applying REST", which is widely used. Semantic interoperation has systems assign the proper/same meaning to data and raw information. An example is interpreting "degrees" as temperature expressed in Celsius.

In-depth or real-world interoperation has systems preserve the intrinsic capabilities to cooperate in their world-of-interest. Indeed, smart systems may be able to communicate perfectly, only to discover that their design causes them to introduce constraints in their world-of-interest that prevent effective cooperation. Poor in-depth interoperability prevents collaborations that intrinsically/physically would have been perfectly possible. Here, domain experts spot the added value that remains out-of-reach solely because of the IT, which is perceived to be overly rigid.

Ideally, in-depth interoperable systems can be integrated into larger systems without the need to "open them up", which among others preserves all warranties and certifications, avoids having to maintain and support numerous variants, etc. It suffices to establish commonplace interoperability to create integrated overall solutions. The main pointer for further discussion can be found in the project proposal. This task/deliverable is the responsibility of UCL. Paul Valckenaers is the key researcher for UCL. The relevant reference from Paul for this task, listed in the proposal, is "On the design of emergent systems: an investigation of integration and interoperability issues" published in "Engineering applications of artificial intelligence". More recent publications from Paul have elaborated in-depth interoperability and its implications further (cf. section 9). This document discusses in-depth interoperability for the smart energy domain.

More specifically, this document reflects lessons learned within the STORY project. Deliverable D3.6 discussed in-depth interoperability reflecting knowledge as it existed at the start of the innovation project. However, the insights had not been translated toward smart energy. A communication gap still needed to be bridged. First of all, the smart energy community still has high hopes (illusions?) that commonplace interoperability will address poor cooperation effectively in spite of the fact that

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646426

Project STORY - H2020-LCE-2014-3

large numbers of talented people have been attempting this for decades with underwhelming results. Here, one of Albert Einstein's famous statements[i] encourages us to look for answers beyond commonplace interoperability.

In contrast to D3.6, this document addresses in-depth interoperability by discussing how to design and implement smart energy components and their composites that have the required and desired properties. *It expands and enhances the concept of a digital twin*.

Normally, the physical twin of a digital twin will be an asset (e.g. a battery). Here, the range of possible physical twins is *expanded by adding activities and mental states* (intentions and policies). Notice that *real-world twin* would be a more accurate wording (than *physical twin*) because its digital twin may mirror non-physical parts of reality.

The twin concept is *enhanced by* converting our digital twins *into blue-collar digital twins;* they participate in the actual monitoring and control (whereas conventional digital twins are white collar twins providing support service in a back office). The solution is intrinsically bottom-up. Highly aggregated systems (e.g. energy markets where highly aggregated energy products are traded as a commodity) are not covered explicitly.

In section 6, this document presents the scientific insights – scientific laws governing what human-created systems can(not) be – that have driven the design of these in-depth interoperability guidelines and concepts. Our blue collar embodied digital twins reflect these insights. Stronger, the insight often define the solution and the proper interpretation of the wordings. The solution based on blue collar digital twins, presented in section 4, minimizes arbitrariness and maximally reflects what is unavoidable. But first, section 3 concisely discusses the contribution or added value of the solution and approach.

3 Lessons learned – smart energy challenges

Participating in the STORY project and witnessing the obstacles that had to be conquered, a number of challenges became apparent, especially when considering large-scale replication and roll-out. In addition, information about such obstacles – and how they have been handled – in older projects (e.g. LINEAR from VITO) has been taken into account. In particular, the overall picture revealed where

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646426

Project STORY - H2020-LCE-2014-3

there was progress and where the state-of-the-art and the state-of-practice did not change. The main surprise was how little had changed between the older projects (LINEAR) and STORY.

The key issue, from a large scale replication perspective, is people. Witnessing and participating in the demonstration implementation activities, the required skills and training for a roll-out of smart energy in large numbers will be problematic, both in finding the right people and in being (un)able to afford them. The issue is less acute at the extremes of the smart energy spectrum. In the TSO domain, personnel costs are dwarfed by investments costs and their systems are small from a complexity perspective. At the other extremity, low-cost commodity devices can be installed and operated with low-skilled personnel (i.e. one size fits all). However, the most lucrative and novel markets, in which customers have the resources to invest (but require smart and tailored solutions), need addressing the personnel issues. Large-scale replication needs:

- Skilled and knowledgeable personnel to have more leverage.
  These scarce assets need to be available for and contribute to much more installations.
- Less skilled, talented, motivated and experienced personnel needs to be 'employable'.
- Proper installation and operations need to be ensured by the above mix of people.
  Today, the more sophisticated installations often are poorly configured/operated and thus fail to contribute to energy savings...
- Skilled personnel must be able to locate each other, as communication through well-meaning managers does not work that well.

Overall, smart energy solutions must not only provide (in-depth) interoperability. They must also render the installations intelligent in manners that reduce the skill and expertise requirements for the people installing, maintaining and operating smart energy installations. Moreover, they need to amplify the contributions of (hard to find) experts across large(r) numbers of installations. And, these experts need communication means that preserve key information. Fortunately, an intelligent approach to deliver in-depth interoperability addresses this.

Moreover, answering this personnel challenge, by rendering the devices and installations more intelligent, brings benefits beyond the personnel matter:

- More installations will be properly designed and built at the first time/attempt.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646426

Project STORY - H2020-LCE-2014-3

- Problems will be detected in digital mock-ups well before building starts.
- Mistakes will be detected during building and before ramp-up.
- Diagnosis and health monitoring will be significantly improved.
- Disputes will benefit from better and more information. They will take less time to be resolved and will often avoid to end up in expensive procedures (e.g. in a court of law).
- Graceful degradation when equipment malfunctions.

In-depth interoperability brings this type of intelligence, generating the above benefits. In addition, it preserves the intrinsic capabilities of the available assets. It is no longer necessary to be aware on beforehand how assets are to be used. Manners to extract added value from the available assets can be implemented later, when more and better information has become available, when the environment has changed, when coordination across organisational borders is needed.

N.B. Concerning commonplace syntactic interoperability, market aspects are dominating; technical solutions are available but not offered. Indeed, competitiveness often depends on the reputation within the energy dimension (e.g. a pump will not break down during 15 years). Then, there is no market pressure to adopt state-of-the-art ICT. To the contrary, teething problems and unfamiliarity may damage a reputation when adopting state-of-the-art ICT. Here, the world needs to change: sticking to legacy ICT has to result in reputation damage. If not, the last meters to sensors and actuators are likely to remain restricted to legacy systems (e.g. RS232C). Migration strategies need to isolate this (e.g. by implementing device drivers that are application-independent).

4 Embodied and Situated Digital Twins implementing the ARTI architecture

This section describes how in-depth interoperable systems can be designed and implemented. It presents a structure to achieve this objective. Alternatives may exist and section 6 delineates the manoeuvring space that is available. However, this section presents what communicates best what in-depth interoperable designs are. Earlier publications, aiming at preserving the whole range of possibilities, revealed to leave too much to the reader. Moreover, they failed to cleanly separate the decision making from the in-depth interoperable elements.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646426

Project STORY - H2020-LCE-2014-3

***A** key concept is the blue-collar embodied digital twin*. The wordings 'blue collar' and 'embodied' have to distinguish our twins from the broadly-known white collar digital twins in support roles. Our digital twin are online, participating in the execution. They are not tools to assist people. When a pump is switched on, its twin is involved. Moreover, embodied twins are a single source of truth for their physical twin. If one control (ensuring a hot water supply) wants the pump to switch on while another control (doesn't want the noise to wake up the baby) wants the pump to stay idle, the twin will see the conflict and provides a service that allows these controls to see it as well.

The concept of a digital twin contributes significantly to communicating what in-depth interoperable designs are about because – intuitively – readers understand that the digital twin will mirror its physical twin and nothing more. It will not be bossy and decide what its physical twin shall (not) do; it will not add something that is not present in the corresponding reality or, conversely, leave out or modify something. Moreover, digital twins are connected through sensors and actuators with their physical twin; they are synchronized in the sense that changes in the world-of-interest are mirrored in the digital world.

In other words, when interoperation experiences a conflict, the digital twins are safe. If they need to change, their physical twin needs to change (and when it does, synchronisation changes the digital twin by definition). Now note that the real world always is in a coherent and consistent state, but not necessarily in a desired state. It suffices for our digital twins to stay synchronized in order to remain conflict-free; they are in-depth interoperable by mirroring real-world consistency and coherence.

Unfortunately, the above is insufficient to develop in-depth interoperable systems and components. In addition, domain expertise is needed and is to be captured in the software artefacts. Moreover, it must be technically and economically feasible for embodied digital twins to "stay connected to their physical twin". E.g. when a digital twin has an aggregated physical twin and this physical twin changes its composition, a monolithic digital twin will struggle to mirror what is happening in the real world. If it is not monolithic, it still needs a compatible structure that allows it to mirror reality (i.e. it needs to mirror the real-world aggregation). This issue is addressed in the next subsections.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646426

Project STORY - H2020-LCE-2014-3

4.1 The ARTI reference architecture – three colours

In smart energy, the reference model for ICT systems is SGAM. It allows, among others, to discuss about the IT systems as such; it resides in ICT space. In contrast and complementary to SGAM, the ARTI architecture remains ICT-agnostic. It addresses the relationship between ICT and the world-of-interest. Among others, it facilitates deciding what the physical twins for our embodied digital twins will be.

The ARTI cube in figure 4.1 distinguishes three colours for software components. *Blue components* are embodied digital twins; they mirror a corresponding reality. *Green components* are *decision making technology* implementations (e.g. mixed-integer programming. A.I. planning systems, or even dispatching rules accompanied by the body knowledge when to use which rule). *Yellow components* implement the decision making, often employing green components.

Blue components remain unaffected by in-depth interoperability challenges; they remain unchanged when the challenge has been addressed. It are the yellow components that have to adapt (usually, some yellow parts need deconstructing to implement the required adaptations). Often, the system needs switching to other green components.

N.B. Expanding or upgrading an application may require additional blue components, necessitate the extending the range of blue components or adding more details and aspects. However, such enhancement always allows to keep offering the existing services and functions with little effort (metaphor: replace tourist map of a city centre with a traffic map of a city and the neighbouring villages; the information services of the tourist map are generated from/included in the traffic map).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646426

Project STORY - H2020-LCE-2014-3

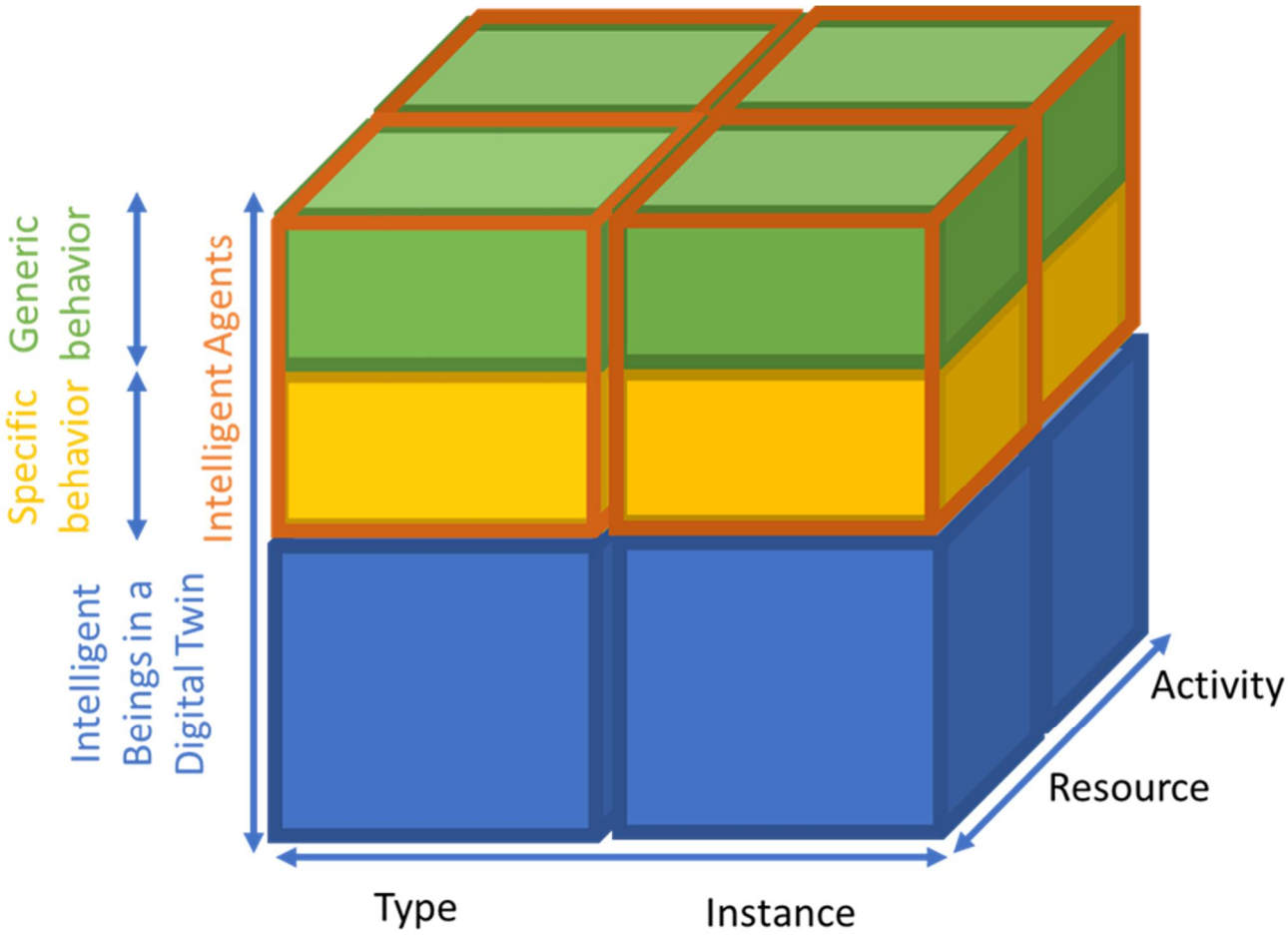

Figure 4.1 – The ARTI cube 2.0

Guidelines related to the ARTI colours:

- Maximise the contribution of blue components. It suffices to establish commonplace interoperability to integrate them without having to deconstruct them.
- Change the collection of green components when indicated. Each green component represents a significant body of knowledge (i.e. multiple person-years); adapting a green component cannot be justified (too much effort) for individual applications. Adapting the services of a green component to a special application is to be handled by yellow components.
- Minimize the inertia of yellow components. Intrinsically and by definition, they introduce constraints in their world-of-interest and risk conflicts.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646426

Project STORY - H2020-LCE-2014-3

- Look for communalities in yellow component that may be addressed by to-be-developed green components (e.g. a socially acceptable behaviour when making commitments) and/or shift responsibilities into to-be-developed blue components.

4.2 The ARTI reference architecture – four types of digital twins

Economic feasibility is the key issue for digital twins in smart energy. Key factors in this matter are:

- Complexity and size of the twin: smaller and simpler means less effort to develop and maintain the digital twin. E.g. the twin of a tube requires less effort than the twin of a wood furnace.
- The longevity and stability of the physical twin. A specific installation may only exists for a few years, perhaps even months. Tubing will be in our smart energy installation for centuries to come. Note how simple and longevity go together.
- The upper bound on user mass determines the ability to spread the required effort over the available sources for funding and for paying what is needed and even desired. Again, simple and longevity are likely to have a larger user mass (to recruit from).
- The user mass diversity. Diversity allows digital twins to become battle-proven in a wide range of conditions. Without diversity, innovative uses of the physical twin are likely to have its digital twin encounter teething problems and poor support (e.g. training materials).

Overall, the targets of our embodied digital twins – the physical twins – ought to be small and simple. Looking at the next section, aggregating components as well as composites thereof (subsystems) will be needed and supported anyhow. This allows to state that errors, when establishing our reference architecture, must avoid physical twins that are too large or complex. “Large and complex” is to be handled by the aggregation, which needs to cope with all such larger physical twins anyway.

Applying the science (cf. section 6) underpinning in-depth interoperability in multiple application domains, the following twin classes emerged (cf. figure 4.1):

- Resource (asset) types.
- Resource (asset) instances.
- Activity types.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 646426

Project STORY - H2020-LCE-2014-3

- Activity instances.

The distinction between types and instances is justified by the radically different nature of their digital twins. Twins of types are (technical) experts. Twins of instances are managers or management assistants. Failure to separate them will decimate their (potential) user masses. Technical processes exist in many organizational arrangements. Similar organizational arrangements share twins of instances, regardless the technical processes therein Conversely, users of a technical process share twins of the types concerned, regardless of the organizational arrangement.

Empirical justification was obtained while applying the underlying science to multiple application domains. When developers were mainly concerned with the managerial aspects, the type became subordinated to instance (e.g. a rigid step list). When developers were mainly concerned with the technical aspects, managing the execution became an afterthought. Positioning types and instances on the same level allows to cover both situations (and has no drawbacks).

The distinction between resources and activities is even more crucial. It equally enlarges the upper bounds on user mass and the diversity therein. More importantly, it renders configuration and reconfiguration to be business as usual, even during execution. Subordinating activities to resources will limit the ICT to expected and anticipated configurations, and it hinders addressing separate concerns with separate activities (e.g. safety from normal operations from monitoring from diagnostics from maintenance). Subordinating resources to activities will be a liability for innovation in the equipment and installations; the activities will only allow what they know. The separation has activities discover the resources (and what they are capable of), whereas resources provide their services (subject to proper allocation).

And as observed already, having four classes while less would be sufficient for smart energy is not a problem at all. The opposite – if additional classes reveal to be needed – may require to re-think and re-invent this document. Note: the available information on SAREF only covers resource types and fails to recognize activities as concept on par with resources. Basic activities are attached to resources (i.e. second-class citizens in software engineering jargon).

To clarify the above further, consider the following examples. As ARTI is about the relationship between relevant reality and the IT (a digital twin) that renders a corresponding reality intelligent

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646426

while maximally avoiding to introduce hard-to-indo real-world limitations, the examples are discussed accordingly. Actual sample source code can be found in D3.7.

1) Resource type example: *tubing* in a thermal installation.

The digital twin of the pipe type delivers services that require technical knowledge such as the geometry or material of its type of tubing. Any instance-specific properties (e.g. quality control data out of the factory that produced the pipe instance) is stored with the instance twin. When instance-specific information (e.g. usage history) is needed to deliver a type-related service (e.g. fatigue issue estimation), this information is retrieved from the instance twin.

An example of a pipe type twin service is to *estimate the resistance that will be experienced by a fluid at a given flowrate*. This is part of the self-modelling capabilities offered by digital twins. This resistance estimate can be provided in a table (with data from testing and/or computation intensive modelling). Any instance-specific information about this resistance will be stored with the instance twin (who does not understand the information) but used by the type twin (who does not keep track of individual instances).

2) Resource type example: *pumps* in a thermal installation.

The digital twin of a pump type provides every pump instance twin with a representation of the supported operations: switch on or off (or select a setting if the pump has variable speed settings) and state queries (e.g. on, off, unknown, broken down, no power, etc.). The type twin knows how to execute those commands, given the proper device driver functionality. It also provides these instance twins with an initial state representation (e.g. unknown).

Information to identify and have access to a physical pump instance resides with its digital twin instance, but this information is passed to the type twin who knows what needs to be done and what the impact on the state information will be. Concerning self-modelling, the type twin is able to estimate the influence exercised on a fluid given the pump instance state (e.g. on or off).

3) Resource type example: *flowmeters* in a thermal installation.

The digital twin of a flowmeter type, similar to pumps, provides every instance with a representation of the supported operations and commands (e.g. perform a self-check). To implement its self-modelling, the type twin may employ the network of twins that can be discovered. When instructed

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646426

Project STORY - H2020-LCE-2014-3

to estimate the flow (rather than read out the device), it may use the fluid instance twin to discover the circuit and query each conduit in the circuit about its estimated impact on the flowrate.

This self-modelling capability may be employed to check system health: a large deviation between the sensor readout and the estimate indicates that something is wrong. Moreover, the self-modelling allows for graceful degradation when sensors fail (and are in locations that make it expensive to replace them). Also, the self-modelling can be executed on digital twins that provide their estimated future state as input (e.g. pumps that are scheduled to function at given time periods). Thus, the flowmeter twins are able to simulate themselves.

In other cases, estimated sensor values can be used to avoid wasting energy. When the application does not need the pumps to switch on whereas a control system requests to measure the fluid temperature (which requires the fluid to circulate), the digital twins may provide the information without energy consumption by the pumps.

4) Resource instance example: *tubing* in a thermal installation.

The digital twin of a pipe within an installation knows the digital twin of its type (cf. above). All service requests on technical matter is delegated to this type twin (e.g. geometry, material, strength). The instance twin contains all state information: twins of connected devices, twin of the fluid inside the pipe (if present) and the twin of the aggregates that contain it (directly).

Notice that it suffices to be acquainted with a single instance twin to discover the entire installation, possibly subject to authorization (e.g. username and password). Applications, similar to web crawlers, may map the system configuration and deliver their service in a model-driven manner.

Furthermore, instance twins cover the entire life cycle of their instance, from installation until removal. They start from 'not yet existing' until 'no longer existing'. When the physical twin instance cease to exist, its digital instance may remain active (as part of the overall system's memory that is readily available), be hibernated (as part of the overall memory that remains available but at a lower cost lower service level) or cease to exist as well (when no trace is to remain available).

5) Resource instance example: a *pump* in a thermal installation.

The pump instance twin extends a pipe instance twin by enriching it state (on, off, unknown, other). A variable-speed pump possesses a state representation that accounts for this. In fact, a generic pump

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646426

Project STORY - H2020-LCE-2014-3

instance twin may remain agnostic on the details of this state representation when it delegates this to its type twin.

6) Resource instance example: a *flowmeter* in a thermal installation.

Again, the flowmeter instance twin extends a pipe instance twin. It extends this pipe twin by referring to its flowmeter type twin, which provides and handles the "measuring services" that can be requested from this instance twin.

The reader may already have detected that the twins of device instances that are pipes with some extra functionality delegate their differences to their type instance. If they manage to do so in full, a single instance twin (software) will be able to serve all of them (and enjoy the combined user mass).

7) Resource instance example: the *liquid* in a circuit in a thermal installation.

The instance twin of the fluid (A) in a given thermal circuit uses the above-mentioned crawling to discover its circuit(s) when it is uploaded. It suffices to make the fluid instance twin (A) acquainted with the instance twin of its entry point (B_1). When loaded into the circuit, the fluid instance twin (A) informs the instance twins (B_1 ... B_n) within the circuit of its presence. In turn, this fluid instance twin (A) permits others to crawl along the devices through which it flows.

The instance twin (A) contains the state of the fluid instance but delegates all technical matter to the digital twin of its type. State information is generated and manipulated by this type twin (T) but stored at and retrieved from the instance twin (A). This state information may comprise temperature (profile), circulation speed, level of turbulence (profile), chemical composition, etc. But the instance twin (A) is able to remain agnostic about such technical matter. It only stores the information, passes it to type twin (T) and replaces it when a type twin (T) has compiled an update.

8) Activity instance example: prevent overheating

Next to normal operations, thermal solar installations need precautions to prevent overheating that would damage equipment. In configurations as shown in figure 5.1, a radiator will be installed to shed excess heat when needed. An activity instance is to be created to prevent overheating.

As with resource types and instances, the exact procedures are provided by the type twin. The instance twin receives instructions to execute. First, it needs to acquire the needed rights over

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646426

Project STORY - H2020-LCE-2014-3

resource instances. This includes ownership over the radiator switch. It can be normal ownership provided the radiator is solely used for overheating prevention.

Next, the activity instance twin needs to be able to ensure that pumps are switched on when needed. But these pumps are to be shared with other activities, in particular the activities that ensure normal operations, which require normal ownership. The overheating prevention acquires pre-emption rights over the pumps. These rights allow the activity to remain invisible until overheating becomes imminent. In the latter situation, these pre-emption right allow it to overrule normal ownership.

Advanced versions may add functionality to nudge, influence and coordinate with those other activities. In particular, advanced implementations supporting embedded predictions allow to recognise overheating on beforehand, and more economic prevention scenarios can be initiated.

9) Activity type example: prevent overheating

Whereas the instance twin (A) is responsible for acquiring the necessary rights over resources, the type twin (B) knows what the available options are to prevent overheating. The type twin (B) provides the instance twin (A) with the required allocations (possibly, multiple options for the instance twin to pick from). When the instance twin (A) reports back (allocations that were successful), Type twin (B) instructs the instance twin (A) to monitor the temperature (again leaving the instance twin to pick from multiple alternatives). And, when the monitoring result indicates imminent overheating, the type twin (B) instructs the instance twin (A) to execute a heat shedding scenario. Here, the instance and type twin execute the NEU protocol [2].

Finally, notice that specialization (inheritance in programming languages) also increases the user mass of digital twins when they share abstract versions of themselves. E.g. a pump instance is a tube instance (offering additional services). The pump instance twin will be an aggregate comprising a tube instance twin and all tube-related services will be provided by the tube instance twin.

4.3 The ARTI reference architecture – Simon's aggregates

As revealed in section 6, real-world systems are created by the aggregation of suitable subsystems. Their sustainability is ensured by replacing such subsystems. And, this is applied recursively until

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646426

Project STORY - H2020-LCE-2014-3

subsystems composed of basic components are encountered. To allow this aggregation to happen and to adapt continuously, subsystems need to integrate well, ideally without deconstruction.

In other words, in-depth interoperability is called for (cf. section 6: to allow integration without deconstruction). Digital twins for aggregates deliver in-depth interoperability. Importantly, they support the time-variant composition of their physical twin, a real-world aggregate. In this manner, the collection of digital twins is able to track its corresponding reality and benefit from its consistency and coherence.

The above is fine provided the aggregates are exclusively composed of blue components. Unfortunately, real-world smart (energy) systems are performers of activities on resources. Performers are aggregates of yellow and blue components (green components are optional). And, deconstruction of a performer into a yellow and blue part often has significant drawbacks (e.g. loss of certification, warranties, the need to retest, retune, etc.).

However, some measures allow these yellow-infected performers to deliver suitable subsystems for aggregation. First, minimizing the inertia for yellow components will minimize the drawbacks for deconstruction. If properly designed, modification of a yellow component has a contained impact; blue components remain valid, certified, tested, and few other yellow components are exposed to the change induced by this modification. Second, yellow components may maximally connect to the remainder of the system through blue components. If a yellow components keeps operating whatever happens with these blue components, in-depth interoperability is ensured, but often with some performance degradation.

Moreover, explicit resource allocation contains the impact of yellow components and their constraint-inducing decisions. When the aggregates manage the allocation of their embedded resources to activities explicitly, a key class of deconstruction scenarios becomes a controlled modification of the aggregate. When a yellow component needs to be neutralized or replaced, *deallocation of resource ownership rights* frees all assets for a new better-adapted composition. If this is mandatory for all activity execution on resources, unforeseen manners of using the available resources become business as usual.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646426

Project STORY - H2020-LCE-2014-3

Furthermore, properly designed blue activity types minimize their requirements for resource allocations; indeed, they must not ask for more than is needed in their world-of-interest. This also renders the aggregates better-suited for integration into a larger system or installation. Note that this minimal demand for resources sometimes will be implemented lazily when the effort to validate alternatives is too big to deliver it until the situation demands it.

To clarify the above further, consider the following examples.

1) Connectors

Pipes, pumps, flowmeters, etc. are connected in a circuit. Their digital twins are aggregates. They comprise connector twins. As connectors and their devices share their life cycle – they live and die together – the device twin need not support time-varying aggregation. In other aggregations, the real-world dynamics are to be reflected.

Connector type twins mirror real-world compatibility. When pipes have a larger diameter on one side, allowing the smaller side to fit inside, and pipes need to be glued together, it is their connector type twin that reflects this. If other pipes have flanges that need to be screwed together, it is the type twin that reflects this, including geometry and material properties.

Connector instance twins reflect their presence within the aggregate instances. Without connectors, (crawler) software cannot distinguish where equipment is connected. Connectors make explicit that devices have specific locations in which they are attached to each other. System discovery by crawling software navigates by querying device twins (D_1) about their connectors (D_1-C_1), connectors about connected connectors (D_1-C_1-C_2) and, finally connected connectors about their devices(D_1-C_1-C_2-D_2). Here, mirroring real-world complicatedness is rewarded by making essential properties observable. Indeed, savant software easily copes with the above, whereas the inverse is not true (i.e. only knowing which device is connected to another and having to intelligently derive what the more detailed topology is).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646426

Project STORY - H2020-LCE-2014-3

2) Activity 112 (in Europe) or 911 (USA).

Based on self-modelling, many resources may detect problems that affect them (e.g. overheating) and activities may know when they may fail to keep up their duties (e.g. heat prevention discovers all temperature sensors are broken). When this happens, their twins call for help; they dial 112 (or 911).

Under 112, an aggregated activity will be registered. This activity comprises triage services feeding service selectors, which forward the call for help to the (hopefully) correct problem handling activity. E.g. resources in the circuit of figure 5.1 know their temperature range. When it is exceeded (or in the advanced systems, when it is predicted to be exceeded), they call 112. Next, this aggregate activity dispatches the call for help to a suitable specific activity and ensures follow-up to make sure adequate help has been provided.

4.4 Self-modelling Twins and Twins of mental states

The examples above reveal that embodied digital twins are executable models of their physical twin. As they are networked in manners that mirror the corresponding reality, an overall system simulation can be constructed with minimal effort. Indeed, when these twins are available the natural manner to conceive and realise a smart energy installation is to prepare it 'in silico'.

The digital twins are used twice: one copy in the smart energy management system (as it would be deployed) and another copy, connected to the first copy, emulating the world-of-interest. The main difference is e.g. that a sensor in the first copy interacts with the sensor in the second copy whereas the sensor in the second copy computes an estimated value when a read-out command is received. These estimations can be made to behave stochastically (e.g. simulating malfunctions and measurement errors). The first copy is aggregated with the yellow and green components in the configuration that is to be deployed. The second copy receives commands (and thus does not need the yellow or green components). It may connect to external (web) services to account for the external environment (e.g. weather). A student assignment[1] realised such a setup already in which a "simulated real system" is managed by a "digital twin system".

[1] https://github.com/SvenBaerten/TAGP_Project_Erlang_Cowboy

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646426

Project STORY - H2020-LCE-2014-3

However, the embodied digital twin approach is able to expand its service further. This requires these twins to mirror mental states of the relevant yellow components. Activity twins mirror the 'intentions' or plans regarding their execution (e.g. when will it switch on a pump, charge a battery) and resource twins mirror their 'policies' (e.g. what happens when activities have conflicting intentions). There is a subtle but crucial difference with including the yellow components. The twins do not impose a choice, they simply observe relevant mental states within their world-of-interest. Note that synchronization is likely to require frequent updating.

When activity and resource twins are aware of relevant mental states in their world of interest, the twin-based management system is able to predict what will happen (for the collective). Thus, it assists in optimizing the system without imposing constraints in its world-of-interest. It counters poor performance by predicting it in time for yellow components to react and respond.

This part is the most futuristic element in the in-depth interoperability story. Further details are out of scope. An example of the underpinning science is the control of system nervousness in which sticking to commitment (a.k.a. socially acceptable behaviour) is managed explicitly.

5 Sample scenarios – added value from in-depth interoperability

Lessons learned from conceiving, realising and operating STORY demo installations reveal that a replication and roll-out on a large scale will face serious personnel issues. Note that these issues may not be noticed at higher levels within organisations. These are issues that, today, are handled (very well) by skilled, talented and experienced personnel. Because they solve their problems, much of it remains invisible within a management hierarchy.

The issue is that this high-quality personnel is needed and, when scaling up, it will neither be available nor affordable. Fundamental to this issue is that the answers and solutions delivered by today's skilled personnel reside in their brains, not in equipment, training guidelines, control software or other software tools. When forced to employ less-qualified personnel, the result will be installations that malfunction, are badly configured, have no effective health monitoring, etc. Projects will suffer serious delays, underperform, and even fail completely.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646426

Project STORY - H2020-LCE-2014-3

Interoperability issues often are the more visible symptoms of the above but other issues will equally cause difficulties whenever skills, talent or experience are lacking. When affordable and widely available personnel is employed to realise smart energy installations in large numbers, they need IT support that either provides whatever is missing or that amplifies the reach of skilled personnel (such that they become affordable) or both. This section presents a number of scenarios in which in-depth interoperability through embodied digital twins provides precisely such a service.

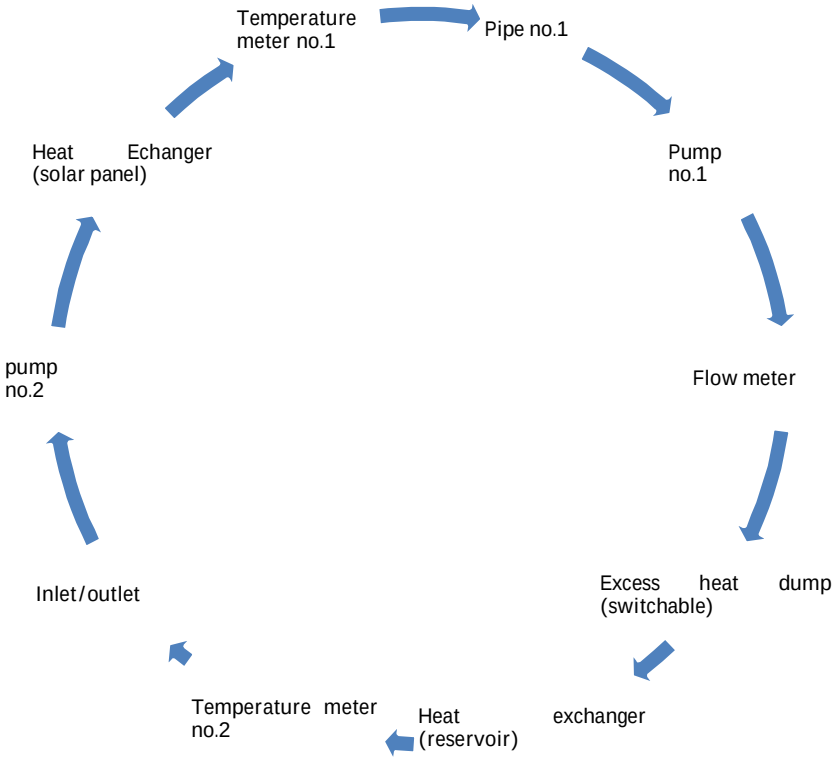

Figure 5.1 – A sample thermal circuit

Consider the sample energy installation in figure 5.1. Inexperienced personnel may overlook the following issues:

- The flow meter is too close to pump no. 1. If this pump is operating, it introduces turbulence in the fluid, which result in faulty flow measurements.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646426

Project STORY - H2020-LCE-2014-3

- Insufficient heat consumption on a hot and sunny day may cause overheating and damage meters and/or equipment.

The self-modelling capabilities of the digital twin will detect the turbulence issue. If the twins are used in the simulation configuration, the issue will be detected in time to reconfigure. If the twin are used during installation, it will be detected while everything still is accessible. If discovered during operations, the twins may learn – from observing and analysing the overall performance – how accurate the flow measurements are and what the systematic errors may be.

Likewise, the self-knowledge of the meters and pumps within their digital twins includes the allowed temperature ranges. Interactions amongst the twins allows the digital twin of the fluid instance to estimate its temperature along the circuit. Accordingly, each twin is able to detect when it risks overheating – in a way, the embodied twins know when their physical twin *experiences pain*.

An emergency service activity will be globally known (registered as process 112 or 911) within the installation. When a digital twin detects that it physical twin is in trouble, this emergency service is informed and a suitable trouble handling activity will be initiated. In the example, the excess heat dump will be switched on and pumps will circulate the fluid to cool it down.

The above scenarios are perhaps anecdotal but they reveal how experience and application domain knowhow is captured in software (in the digital twins). It is no longer necessary to employ personnel that knows everything that needs to be accounted for. Importantly, the software delivers its service and support for a wide range of systems and situations. The safety-ensuring services allow for more adventurous customized installations when first-time variations and configurations may experience teething problems affecting their performance but not system integrity.

In the advanced installations, which have digital twins mirroring mental states as well as self-modelling, the above emergency situation (overheating) will be predicted. When such issues are discovered on beforehand, more economic measures may address the issue. E.g. overheating may be avoided in manners that still make good use of the available heat.

Overall, the embodied twins allow to monitor an installation, perform diagnosis, and learn about a system. The latter may prove useful when sensors malfunction, i.e. to detect malfunctioning and make the correct diagnosis (= assign an inconsistency to the correct cause), and even to compensate

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646426

Project STORY - H2020-LCE-2014-3

a malfunction by a virtual equivalent (= self-modelling – used to monitor system health by comparing actual measurements with model-based estimates – starts to offer its estimates as a replacement for the faulty measuring equipment).

Overall, the embodied digital twins capture domain knowledge such that personnel may cover less and still produces and operates advanced smart energy installations. It addresses the scarcity and affordability of personnel; it offers jobs to a wider range of people. At the same time, it prevents problems in installations (less delays, less repairs, less costly mistakes, better tuning and performance). It makes installations more observable. The latter results in less conflicts and less costly conflict resolutions (e.g. avoid going to a court of law). In other words, designing ICT systems such that they are in-depth interoperable renders them self-aware and self-knowledgeable without hardwiring specific configurations. This brings more benefits than only interoperability.

Finally, note how embodied digital twins are well-suited to take advantage of advanced technologies such as installers wearing smart glasses in combination with equipment that is tagged with QR-codes. Such a set-up will allow the twins to observe what is installed, checking it against the installation plans en ensuring that installation plans and documentation are correct.

6 Scientific laws of the Artificial: the science underpinning the solution.

Readers who are willing to accept the proposed solution in section 4 and have no desire in understanding or in challenging why in-depth interoperability requires to adopt what is discussed in section 4, may skip this section. Readers who are not prepared to accept section 4 on its own authority shall read this section. The guidelines and solutions presented in section 4 minimize their arbitrariness. To a significant extent, they follow unavoidably from the underlying assumptions that are discussed hereafter.

Assumption 1: Bounded Rationality.

Nobel Prize winner H. Simon drew our attention to the existence of *scientific laws of the artificial,* governing manmade artefacts in the same manner as Newton's laws of physics or Carnot's Principles reflect unavoidable aspects of the physical world. When the proper preconditions apply, ignoring these laws is not an option.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646426

Project STORY - H2020-LCE-2014-3

Einstein's laws or quantum physics are more accurate metaphors for Simon's laws of the artificial: they only become relevant in domains that are less familiar in everyday life (i.e. respectively at velocities approaching the speed of light or at below-microscopic sizes). Simon's laws start from bounded rationality: *human intelligence is unable to design and implement the prevailing successful large systems rationally/intelligently*. Much has to emerge without any human organisation ever devising and building it. There is no "intelligent design" and adding more brains to a team creates more communication and coordination burden than those additional brains contribute. Here, Simon is outside the more familiar setting in which single organisations, especially large ones, are managing their own business.

But, it is reasonable to state that the ambitions of the smart energy community stretch well into this domain where bounded rationality is relevant and often dominant. Even the largest organisations cannot (micro)manage what is envisaged or needed. At the top level, systems emerge, thrive, survive and perish without any organisation at the steering wheel. *Organizations,* even the largest ones, *are manoeuvring intelligently within their environment, but they don't design their environment*. Simon was among the first to spot that there are fundamental patterns in this domain: laws of the artificial.

Assumption 2: Artificial Narrow Intelligence

Quoting Karina Vold, University of Cambridge, in the proceedings of 1st HUMAINT workshop (cf. JRC Conferences and Workshop reports):

> *While there have been many astonishing feats by AI in the last decade—even in just the last year—there continue to be many fundamental differences between human and machine intelligence. For one, many of the headline-making accomplishments by AI have been in highly specialized domains, or in what experts call Artificial Narrow Intelligence (ANI).*

It is safe to state that innovation – not waiting for still-to-be-achieved breakthroughs in basic science to mature – may assume that any kind of computer-basic intelligence will be delivering narrow intelligence, even when employing cognitive acrobatics from artificial intelligence.

In a way, software will be savant. It brings hyper-calculus, photographic memory, honours requests within milliseconds, etc. but it has no common sense, no general intelligence and rarely knows what

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646426

Project STORY - H2020-LCE-2014-3

is motivating its actions. It is autistic in the sense that it struggles to filter relevant information out of all the stimuli it receives. *Computer-based intelligence needs to include within its narrow focus everything it needs to deliver*. One cannot expect any kind of robustness in this respect.

Summarizing the above assumptions, humans are limited in how much information processing they can perform in a given amount of time, even with unlimited resources, and computers are limited in the scope of the concerns that they may need to address. When we add finite time windows to produce results and a competitive environment, patterns emerge that severely limit the kind of artefacts that may emerge and persist. Humans need to build sufficiently large systems fast. Computers need to select their targets carefully. Disregarding the implications of these assumptions, humans produce solutions that are no longer needed or software lacks vital skills/features.

6.1 Simon's law : time-varying aggregation hierarchies

Simon's key insight emphasizes the omnipresence of aggregation hierarchies in our world. Almost nothing has a monolithic structure composed solely of basic elements. There invariably is a recognisable structure. The human body is composed of organs, organs are composed of … Society has families, communities, nations, … Moreover, the composition of these aggregates varies over time.

Simon uses a parable to make this observation plausible:

> *There once were two watchmakers, named Hora and Tempus, who made very fine watches. The phones in their workshops rang frequently and new customers were constantly calling them. However, Hora prospered while Tempus became poorer and poorer. In the end, Tempus lost his shop. What was the reason behind this?*
>
> *The watches consisted of about 1000 parts each. The watches that Tempus made were designed such that, when he had to put down a partly assembled watch, it immediately fell into pieces and had to be reassembled from the basic elements. Hora had designed his watches so that he could put together sub-assemblies of about ten components each, and each sub-assembly could be put down without falling apart. Ten of these sub-assemblies could be put together to make a larger sub-assembly, and ten of the larger sub-assemblies constituted the whole watch.*

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646426

Project STORY - H2020-LCE-2014-3

When Tempus had to answer the phone, ringing regularly, he had to put down the not-yet-finished watch and start over from scratch after answering the call. In contrast, Hora had to redo less than five assembly steps on average after answering a phone call.

Systems in a competitive environment grow larger as long as this brings superior competitiveness. But in a dynamic environment, only a limited time window will be available to grow larger. Systems that aggregate large(r) subsystems are able to emerge faster (i.e. within small time windows) and adapt faster by replacing entire subsystems. This is applied recursively and Simon's observation of how our world is structured became plausible.

For in-depth interoperability, Simon's law implies that composition aggregation, with ***a*** *time-varying membership, needs to be part of the supported reference/system architecture*. Building large systems from scratch while only using basic components is not an option. Rigid (control) hierarchies are also not an option; each snapshot may be hierarchical but the structure will evolve over time. Moreover, components and subsystems may belong to multiple hierarchies depending on the circumstances (e.g. private life versus work environment). Smart energy appliances and systems cannot assume a predetermined or static environment.

6.2 Complex-adaptive systems theory: autocatalytic sets or critical user mass

A second insight in non-trivial systems that emerge without intelligent design was discovered by researchers in the domain of complex-adaptive systems. The mechanism was discovered while assessing how plausible the *emergence of life by chance from basic organic material* could be. The naïve version of such a theory proofed to be extremely improbable; the probability of creating an amoeba by chance within the life span of our universe was almost zero. Some mechanism had to improve this drastically.

The answer (or at least a significant improvement) was the autocatalytic set. The naïve version assumed that random combinations, occurring when e.g. lightning injects energy into a pool of basic organic material, had to result in some of the smallest known forms of life. Autocatalysis allows for an undetermined number of intermediate forms. And, Simon's law reveals that intermediate forms composed of intermediate forms are likely to dominate anyhow.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646426

Project STORY - H2020-LCE-2014-3

With autocatalysis, the random combination only needs to produce coarser materials that are catalysts for themselves and that are sufficiently stable to survive until the next energy injections allow them to (re)produce more of themselves. Note that is an exponential growth and likely to consume most of the materials that are used for creating more members of such autocatalytic sets.

Here, a side-effect of autocatalysis is 'lock-in' into the first set that grows beyond a critical level. It rapidly consumes most of the available materials and, thus, denies alternative sets from emerging. It becomes a monopoly unless other mechanisms and systems dynamics prevent this. Among others, this causes all those legacy issues that are denying our society a lot of value and savings.

Similar to Simon's laws, there is abundant empirical evidence. All forms of life (fauna and flora) are autocatalytic; they reproduce. Most autocatalytic sets in nature have two members: male and female. Moreover, within this landscape, greater variety has emerged (e.g. symbiotic, parasitic). Often, an ability to adapt and evolve has become part of the autocatalytic capability when reproduction needs to fit in a world that changes continuously. However, there are no autocatalytic sets made from human-made artefacts.

Nevertheless, non-trivial artefacts participate in (very strong) autocatalytic processes. The sets to which they belong contain their users and developers. It is their (critical) user mass that brings the self-reinforcing of their presence in our world. Users bring the (economic) resources and the information (feedback) that enable our (software) artefact to be successful and remain successful. Note that a more sophisticated artefact will need a larger user mass. Again, empirical evidence is abundant (e.g. Apple, Microsoft, Google, or Java, C/C++).

For in-depth interoperability, we see a world in which the dominating non-trivial elements are members of autocatalytic sets. Hence, it is important to facilitate membership of strong autocatalytic sets for the software and other artefacts that we develop. Here, the design of a software/artefact does not determine its user mass. It rather determines an upper bound for this user mass. *It is necessary to maximise the upper bound for user mass,* especially relative to artefact's sophistication.

Simon's aggregation will occur in a world that is dominated by members of autocatalytic sets. Indeed, that are the systems and components that are widely available, sustainable, supported, etc. Lastly, there is some bad news for in-depth interoperability, which is key for aggregation as seen by Simon.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646426

Project STORY - H2020-LCE-2014-3

The above-mentioned lock-in allows autocatalytic sets to absorb valuable assets (including user and developer communities) for artefacts that inherently are unsuited for further integration. They are dead-ends that need deconstruction for in-depth interoperation. It is this lock-in phenomenon that makes (platform) disruptions to have such a drastic impacts while the ensuing progress is immense.

6.3 Design for the unexpected: compose-ability as a mandatory requirement

The first two insights follow from bounded rationality (a.k.a. intelligent design is impossible). This insight follows from narrow intelligence: in-depth interoperability has to be a top concern for the designers and developers of a software artefact. It is an illusion to expect a heavily-savant intelligence to exhibit the flexibility that is needed for integrate-ability when it is not considered to be a mandatory requirement from the beginning. Human-like *general artificial intelligence* still is in an embryonic stage within very fundamental/academic research.

In-depth interoperability or the ability to integrate without having to 'open up or deconstruct' is not about the ICT (that is commonplace interoperability). In-depth interoperability looks at the relationship between IT and a corresponding part with reality (the world-of-interest). It has two rules:

- First of all, do not add limitations to the world-of-interest.
  Do not hardwire choices or decisions in the world-of-interest.
  Preserve intrinsic capabilities, minimize consumption/ownership.
  Only demand what is really needed. E.g. resource allocation for assets, embedded in an aggregated system, is a mandatory feature.
  Note that a digital twin, mirroring its physical twin, fits this well.
- Secondly, when it is inevitable to impose limitations, minimize their inertia.
  The decision-making is to be isolated in separate components or containers.
  The decision-making is considered unsuited as a basis on which to build other parts.
  When interoperation or integration requires to deconstruct such a decision-making component, there shall not be a cascade of other components that need deconstruction and adaptation as well. In contrast, it is safe to rely/build on the decision-free components.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646426

Project STORY - H2020-LCE-2014-3

To get a better idea, on-line maps are decision-free; they simply mirror a corresponding reality. When integrating a map, conflicts – in the real-world dimension of the problem – are not solved by modifying the map. Moreover, when a map is upgraded, all the existing services can easily be preserved (e.g. computed from more detailed information). A route description is not decision-free. When the envisaged route encounters an obstacle, our savant intelligence will struggle.

Section 4 introduces the embodied digital twin. The architecture for systems that adopt this kind of twin comply with these rules (and account for the two other insights above). The reader is referred to section 4 for further discussion of in-depth interoperability and design for the unexpected.

Note however that this compliance imposes a different burden-of-proof on the designer: the design of software artefact has to be such that it survives conflicts without deconstruction. The main mechanism to obtain that is to be shielded by a corresponding reality; anything in conflict with the artefact is also in conflict with reality. A key contribution of adopting section 4 is that the software artefact will be able to cope with a dynamic reality in which change is omnipresent and never-ending.

6.4 Collective imagination: self-modelling in the skill selection

This part may be optional/absent in numerous smart energy installations. However, the more sophisticated ones will include planning and/or optimization functionality. Here, there appears to be a conflict with the previous insight: planning and optimization inherently concerns decision making. They impose choice within the world-of-interest and they are instrumental for good performance. How can this coexist with in-depth interoperable components (such as digital twins in section 4)?

The key element, shared with planners and optimizers, is the ability to imagine what will happen. For in-depth interoperability, this has to be a collective ability to imagine what will happen (when all systems and components are interoperating). To preserve in-depth interoperability, the choices in the world-of-interests have to be treated as 'externally given'. These are plans, conflict resolution policies, etc. that may change (frequently) and nothing more. They are 'mental states' that can be mirrored by their own digital twin. Interoperable parts never confuse plans, intentions or policies with reality; they are mental states within a world-of-interest and nothing more.

The combination of self-modelling provided by digital twin of assets and activities with twins reflecting mental states allows for 'embedded simulation' generating an image of the collective future

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646426

Project STORY - H2020-LCE-2014-3

that is to be expected. It influences the decision making exclusively by computing what will happen while positioning the decision making in the same space as the assets and activities. Poor decisions will be countered by "revealing their expected poor performance in time" but without any interference beyond that. *Intelligent in-depth interoperability needs self-modelling to be among the mandatory requirements for (advanced) in-depth interoperable components and systems*.

6.5 Remarks

This document only discusses insights that are relevant for in-depth interoperability. There exist more insights of scientific laws that are relevant for e.g. control structures or continuous improvement. Indeed, conventional hierarchical control is unable to cover from top to bottom and the autocatalytic sets comprising users and developers may evolve analogous to biological systems. Importantly, science has not (yet) produced a robust answer to the problems caused by 'lock-in'. It explains why we often need disruptions to make progress but it does not put forward smoother routes out of a lock-in's negative effects (e.g. how to phase out legacy systems). Perhaps, the inertia limiting in 6.3 may improve this situation in future.

This section 6 is not necessary to understand the remainder of this document. However, when something remains unclear in section 4, it is the underlying science that determines the correct interpretation.

7 Conclusions

This document provides insights and guidelines concerning *in-depth interoperability* for the development activities within the STORY project. It introduces *blue collar embodied digital twins*.

In-depth interoperable systems and components *capture application-domain knowledge* into digital twins of resources and activities. This brings benefits beyond interoperability. It allows to recruit personnel from a broader range of skill levels and still have correct-functioning and well-performing smart energy installations. It allows to ensure system health and integrity by separate subsystems, facilitating innovative installations. It allows advanced installations to predict collective behaviour, which allows for optimizations (without centralization). The lessons learned from participating in STORY point to these other-than-interoperability services as instrumental for large-scale roll-outs.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646426

Project STORY - H2020-LCE-2014-3

This document discusses how to achieve in-depth interoperability. A lot of related matter remains to be discussed. To achieve in-depth interoperability, as discussed, a small semantic gap between the software tools and reality is desirable. This requires a lot of people to leave their comfort zone when "active computing processes" are key elements in closing this semantic gap. Furthermore, less technical matter such as privacy are relevant and important. Conceptually, embodied digital twins open up possibilities when sensitive information reside with its twin, which places information with its rightful owner. Finally, sample implementations of embodied digital twins (i.e. software source code) is provided and documented in D3.7.

8 Acronyms and terms

SGAM	Smart Grid Architectural Model	ICT-centric reference architecture
ARTI	Activity-Resource-Type-Instance	Reality-centric reference architecture
NEU	Next-Execute-Update protocol	Type-Instance interaction model

9 References

[1] Valckenaers, Paul, Van Brussel, Hendrik. *Design for the Unexpected: From Holonic Manufacturing Systems towards a Humane Mechatronic Society*. Butterworth-Heinemann[2], November 2015.

[2] Valckenaers P., De Mazière P.A. (2015) Interacting Holons in Evolvable Execution Systems: The NEU Protocol. In: Mařík V., Schirrmann A., Trentesaux D., Vrba P. (eds) Industrial Applications of Holonic and Multi-Agent Systems. HoloMAS 2015. Lecture Notes in Computer Science, vol 9266. Springer.

[i] Insanity: doing the same thing over and over again and expecting different results.

[2] store.elsevier.com/product.jsp?isbn=9780128036624

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646426

Project STORY - H2020-LCE-2014-3