Developing the Systemic Digital Twin of an Industrial Enterprise with Σ

Daniel Krob[1] and Antoine Rauzy[2]

Abstract

This paper presents our approach to enterprise systemic digital twins, that is to say digital twins of industrial processes based on a systemic approach, and the semantic fundamentals of the formal system specification language Σ (Sigma), which synthesizes systemic, stochastic, object-oriented, synchronous and parallel modeling paradigms, on which it relies. We especially propose a standard methodology for designing, constructing and validating a systemic digital twin, which is illustrated in details on a case study, namely the industrial transformation of a port. We provide a short conclusion highlighting the key business benefits that are obtained in this way, the most important one being a facilitated development and evolutive maintenance for a systemic digital twin.

Keywords

Design, Digital twin, Formal language, Model-Based Systems Engineering, Simulation, System.

[1] Corresponding author – Address: Systemic Intelligence Group – 10, rue de Penthièvre – 75008 Paris – France – Email: daniel.krob@systemic-intelligence.net

[2] Address: Norges teknisk-naturvitenskapelige universitet, Fakultet for ingeniørvitenskap – Verkstedteknisk, P307, Gløshaugen – Richard Birkelands vei 2B – 7491 Trondheim – Norway – email: antoine.rauzy@ntnu.no

0. Introduction

In a globalized and highly competitive economic environment, the reduction of time-to-market and the improvement of the performance of design & manufacturing processes for complex industrial products have led, under the founding impetus of Grieves and Vickers [1-2-3], to the emergence and progressive dissemination throughout industry over the last twenty years, of sophisticated digital tools, called *digital twins* [4-5-6], in order to more effectively equip industrial activities and assist in optimal decision-making in industrial contexts [7-8]. A vast research movement has therefore emerged and developed significantly over the last ten years around digital twins for industry, with objective to properly establish their conceptual foundations [2-7-8-9], to propose methodological frameworks for their design [7-8-10] and to specify their scope of industrial application [11-12-13].

Let us now recall that a *digital twin* is a virtual representation of a technical or socio-technical system, synchronized with the actual system through data exchanges, that serves as the digital counterpart of a concrete product – when the system of interest is such a product – or of a business process – when the system of interest is an enterprise – which can be used to simulate, analyze, predict or optimize the evolution of the considered industrial system among time. Note that an enterprise shall be taken here in the meaning of enterprise architecture [14], which refers to any set of hardware, software and/or human resources, put altogether to achieve some common mission. This is a quite large concept which captures in practice any part of a company.

Today, digital twins are integrated digital tools that allow continuous improvement of a given product or process from a global and end-to-end perspective. In practice, they mainly belong to the family of, either computer aided design tools (CAD), focusing on geometrical representations of a system, or model-based systems engineering tools (MBSE), dealing with functional-oriented models of a system. But, these approaches are not providing the full potential of a "real" digital twin: first of all, simulation and predictive capabilities of CAD and MBSE tools are still poor, which limits their business benefits; moreover, it is also difficult to connect the geometrical & functional representations of a system (see Figure 1), since these two paradigms were just not thought to work together, which leads to siloed geometrical & functional digital twins, at the opposite of the idea underlying this concept.

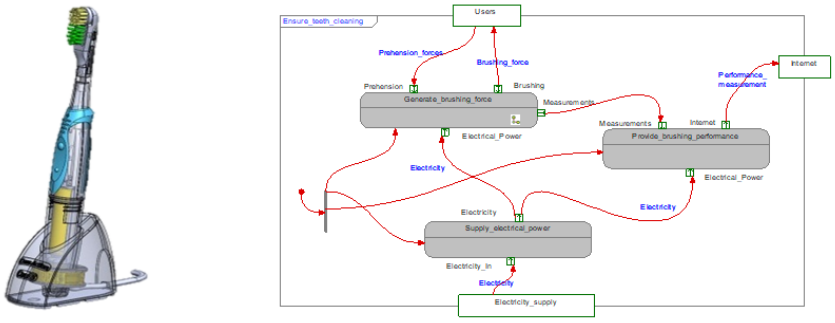


Figure 1 – Geometric and functional representations of a given system, here an electrical toothbrush

We think that this situation is mainly coming from the fact that geometry is usually the starting point of the current digital twin approaches among the industry. Geometry is indeed well suited to support physical simulations related to various types of physical interactions with a given system. But geometry is quite badly adapted when one wants to deal with cost, quality, delay and/or technical or business performances of a system, that can be much more easily addressed by a functional approach, which moreover could also be an entry point to physical simulations based on geometrical representations.

This explains why our approach to enterprise digital twins, as presented here, is based on functional modeling fundamentals since we do believe that this is the only right way to deal with digital twins.

However, most of the current functional-based tools are not satisfactory at all from a digital twin perspective, since they just lack simulation capabilities. This is fundamentally due to the fact that these tools are based on purely syntactic descriptions of a system, without any underlying formal semantics, as defined in mathematics [15]: one indeed knows since a long time that simulating in an unambiguous way a syntactic description of a system requires a mathematical equivalence between a denotational and an operational semantics for such a system [15]. The lack of such semantic foundations jeopardizes most of the existing functional-oriented approaches towards digital twins. This is why we decided to avoid this problem by choosing an underlying S2ML + X formal approach which has already proved its value in the system safety domain [16-17], to deal with functional digital twins, though not visible from the end-user in order to remain as pragmatic as possible, exactly as software programmers do not have to know (formal) semantics of programming languages [15] to do software programming.

We also think that it is also crucial, especially when dealing with enterprise digital twins which is the specific focus covered by this paper and by the Σ approach that we are developing, to take uncertainty into account as a core feature of any enterprise. Any variable naturally associated with an enterprise, for instance the time required to manufacture a product or the number of products produced per year, has indeed not a fixed value, but shall rather be considered as a random variable with some specific probability distribution (see for instance Figure 2). This explains why we chose a stochastic modeling approach for our enterprise digital twins which can be seen as an adaptation in this new context of the classical similar approaches used in system safety [18] as already mentioned.

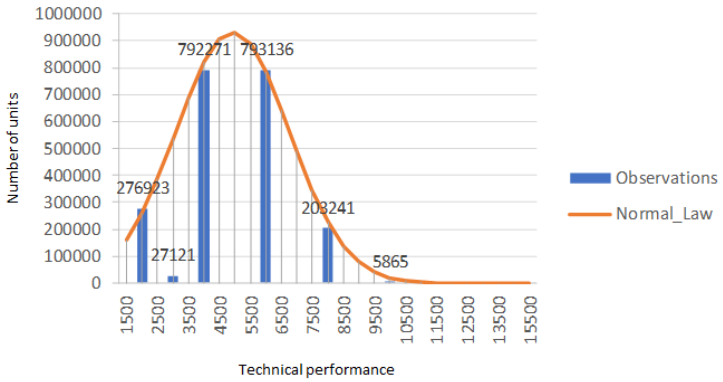


Figure 2 – Example of a Normal probabilistic modeling of the number of products sold per year by an industrial company (each blue bar corresponding to a different type of product of a same family)

The above discussion on uncertainties leads to two fundamental features of an enterprise digital twin, as we envisioned it. First, it consists not only of a model of the considered system, or a series of models, but also of operational data. In this last matter, we propose here a Bayesian approach: at first, data may be scarce or relying on expert judgement for the most part. These preliminary data can then be refined by feedbacks from experience, resulting from system operations. The digital twin also aims at being continuously improved along the life cycle of a given industrial system: strictly speaking, it learns from operations. Artificial intelligence techniques can be especially used to support the extraction of relevant indicators as well as, more technically, during simulations. In a second direction, a digital twin does not aim at predicting "the" actual future of the system. Rather, it is a way to explore its possible evolutions. With that respect, we believe that decision making in complex socio-technical systems has much to do with risk analysis: decision makers need to assess possible scenarios of evolution (at least the main ones), to evaluate their likelihood and their potential impacts on enterprise operations, and

eventually make decisions with such digital twin-based information. A digital twin is then an advanced tool to support strategic decisions making, and to adjust a business strategy according to the dynamic evolution of the considered system environment.

This paper intends therefore to present our approach to digital twins which integrates all core features that we just sketched here above. It is organized in three main sections, which are respectively dedicated to the specific principles of the so-called systemic digital twin approach that we are following, the fundamentals of the formal system specific language Σ on which it relies, and to the description of the systemic digital twin design, construction and validation process that we propose, followed by a short conclusion highlighting the key benefits that are obtained in this way.

1. Principles of systemic digital twins

We already discussed some key features of our digital twin philosophy, that is to say the functional-based approach with formal fundamentals and the stochastic modeling paradigm that we propose. A last key feature of our approach is the systemic vision on which it relies, as popularized by Sterman [19] in the context of enterprise modeling [14], that we shall now present.

1.1 Systemic digital twins

As already stated, we are indeed specifically proposing a *systemic digital twin* approach for enterprise modeling and simuling, a core concept that can be traced back to our seminal paper dedicated to the systemic modeling, simulation & optimization of the covid-19 crisis response [20]. This means that, contrarily to many digital twin toolsets, we are not interested by principle to construct a perfect detailed digital clone of a given industrial enterprise: we rather want to propose adequate digital tooling to allow deciders to take their most important strategic and/or operational decisions, for instance choosing the optimal way of working for engineering, supply chain sourcing & procurement, manufacturing, maintenance & services or industrial operations, which can usually not be optimized locally, but request a global end-to-end approach.

This is why we are speaking of systemic digital twins, since to address these typical issues, one needs to take a systemic approach, as classically proposed by systems engineering (see [21-22-23-24]), but applied here in enterprise contexts rather than in product development contexts. As already stated, such an approach was especially proposed by the MIT group of Sterman, as synthesized in his famous textbook on business dynamics [19] (see for instance Figure 3).

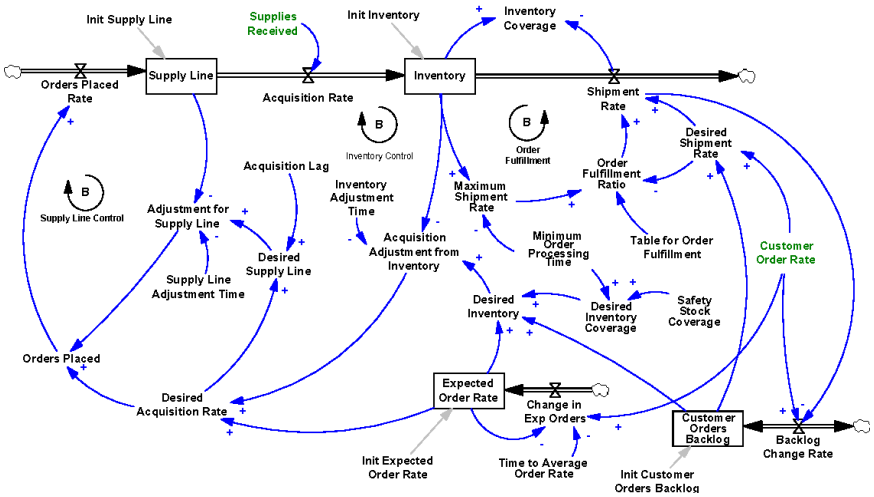


Figure 3 – Example of a systemic model, here for modeling a supply-chain strategy, proposed by Sterman [19]

While the philosophy of our systemic digital twin approach is the same than Sterman's one, the main difference with this last approach stands in the underlying mathematical modeling of the enterprise system behaviors. Sterman uses only differential equations in this matter, when we do not think that it is a good choice for two main reasons: the first one is that enterprises are fundamentally discrete events systems whose behaviors are badly captured by continuous modeling, the second one is that differential equations are generally speaking very sensitive to small variations of initial conditions and so they are not the best tool to simulate accurately complex enterprise behaviors. This is why we choose here a discrete approach to enterprise modeling since it seems both more realistic and more accurate in terms of simulation. Note also that, technically, discrete event simulations are also much less computational resources consuming than solving differential equations as it makes it possible to let the time progress with long jumps rather than with small time intervals. Finally, it is much simpler to develop good semantic fundamentals with this approach since classical frameworks and results [15] can then be more easily adapted in such a framework.

1.2 WorldLab and Σ (Sigma)

It is now time to introduce WorldLab and Σ (Sigma) which are the software tools on which our systemic digital twin approach is based in practice. First of all, WorldLab is the name of the software environment that we are using for managing systemic digital twins. It can be seen as equivalent to a software engineering workshop for enterprise models instead of software programs. WorldLab indeed allows both to model an enterprise, based on the specific Σ system modeling language that we will introduce soon, and to generate automatically the associated systemic digital twin – based on a compiling approach – that can imulate & analyze the underlying Σ-based enterprise models.

The WorldLab technology has two facets dedicated to two types of users: 1) WorldLab Workshop is first a systemic modeling & simulation workshop where an expert engineer can model an industrial system, taking into account its functional behavior and its data, with our formal system specification language Σ and develop the associated systemic digital twin, 2) WorldLab Player, automatically generated with WorldLab Workshop, is secondly the experimentation interface which offers a concrete systemic digital twin to the business users which can, with this business tool, simulate an industrial system, evaluate business indicators and compare & optimize industrial scenarios for the modeled system (see Figure 4). WorldLab shall therefore be seen as a tool suite dedicated to build systemic digital twins. At this point, note also that WorldLab is based on the same core principles than the 20-year-old AltaRica model-based safety technology (see [17-18]) from which it inherits both a number of key features and its industrial robustness.

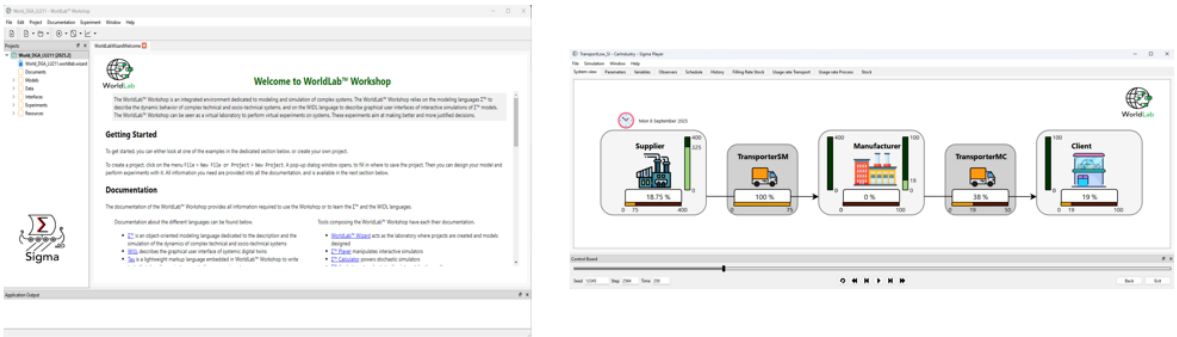


Figure 4 – The two sides of WorldLab: WorldLab Workshop (on the left) and WorldLab Player (on the right)

The core part of WorldLab is of course the Σ language which is presented in details in the next section of this paper (see also [25]). We shall point out here that Σ is a modeling language especially dedicated

to systems dynamics, i.e., to the simulation of dynamic evolution of technical and socio-technical systems, with a special focus on enterprises. Although it involves several concepts borrowed from object-oriented programming languages such as C++ or Python, and prototype-oriented programming languages such as JavaScript, Σ differs from these languages in two fundamental respects:

- first, it is used to design models and not programs: once designed, models can be assessed by means of various tools, notably by means of step-by-step or stochastic simulation;
- second, conversely to programming languages, it embeds the notion of time and distinguishes instructions that take no time, and processes that take an explicitly specified time.

In this last matter, Σ was also inspired by synchronous languages which are managing sequences of data among time [26]. The main difference however comes from the fact that Σ can manage random variables distributed according to explicit probability distributions and also time series associated with arbitrary time slots. Note moreover that, eventually, a model in Σ describes a set of processes running in parallel, and modifying data structures that represent the state of the system under study. This last property especially allows to model deformable systems whose structure may evolve along time.

One shall finally point out the main core difference between programming and our modeling approach: even if Σ looks like a programming language, it is fundamentally different, the main difference coming from a pragmatic requirement which does not exist in programming: the key point when designing a model in Σ is indeed to *capture reality* as well as possible, which is totally specific to modeling and moreover cannot be really formalized. Modeling remains indeed an art …

1.3 Systemic digital twin development process

Developing the systemic digital twin of an enterprise with the support of WorldLab and Σ can be achieved through a standard process that we shall now quickly present and that will be discussed more in details in a forthcoming section (see Figure 5).

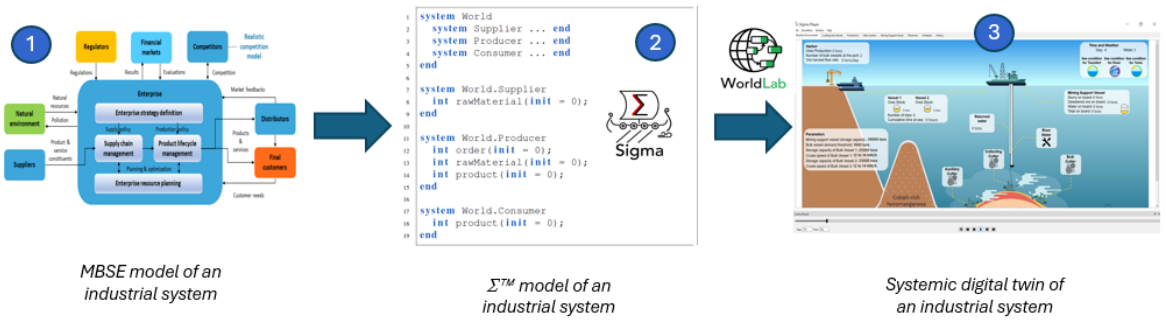


Figure 5 – Overview of the systemic digital twin development process

This systemic digital twin development process has three main phases, respectively focused on designing, constructing, and validating a systemic digital twin associated with a given enterprise, whose contents are sketched here below:

- *Phase 1 – design*: this first phase intends to prepare the core foundations on which a systemic digital twin shall be built: it starts by clarifying the business issues that the targeted systemic digital twin intends to solve, which is key since this initial step will provide the core orientation to give to a systemic digital twin, followed by identifying the enterprise system scope that shall be analyzed through a functional systemic enterprise model, that will be later be translated in the Σ language, when totally stabilized;

- *Phase 2 – construction*: the second phase consists then in technically developing a model using the Σ language within the WorldLab platform: one shall here use the material provided by the first phase in order to specify the key systemic variables & data that will be manipulated in Σ; in a second step, one shall develop the model in Σ for the considered enterprise, which form the kernel of the systemic digital twin, before creating the relevant enterprise decision-support dashboards & alerts that shall help to solve the initially identified business problems;
- *Phase 3 – validation*: the last phase is finally a business phase where one shall validate the systemic digital twin by creating & simulating representative enterprise evolution scenarios, then analyzing the associated simulation results & alerts provided by the systemic digital twin and proposing on that basis relevant business recommendations. At the very end, a last step is dedicated to installing the systemic digital twin as an operational tool in the organization, leading to continuous business improvement through regular evolution scenario creations and associated simulations, analyses & alerts, and possibly revisions of the underlying Σ model.

A synthetic presentation of this process can also be found in Figure 5. Note finally that section 3 is eventually dedicated to a step-by-step detailed visit of this process on a case study, which explains why we are not eliciting it more in details here.

1.4 Systemic digital twins in practice

To conclude this section, let us highlight how systemic digital twins are currently applied in practice to enhance decision-making processes in complex industrial operations. It is important to note that systemic digital twins are not primarily designed for real-time decision-making, unlike many existing digital twins. Instead, they serve as tools to support strategic or operational decisions within business time, meaning that they align with the standard business flow of an industrial company's governance and decision-making processes.

The first key point is of course *virtual-real interaction*, which we achieve here through a configuration mechanism. Using a systemic digital twin, industrial architects can indeed define industrial scenarios via a scenario configurator, which allows them to set and select a range of free parameters related to an industrial system of interest. They can then simulate & evaluate these scenarios against predefined key performance indicators (KPIs). This process enables industrial architects to understand how the real system behaves and evolves over time under the considered scenarios. As a result, they can refine or modify the scenarios to identify more relevant options aligned with specific business objectives. In many concrete cases, this mode of interaction is entirely sufficient for industrial purposes: industrial stakeholders often do indeed not seek the "optimal" scenario, which, in any case, may be meaningless due to the multi-criteria nature of such optimization problems. Instead, they aim for a relevant and reasonably effective scenario based on given business indicators.

A second key point is *optimization guidance*. While the most common practical approach to industrial optimization with systemic digital twins involves comparing pre-defined industrial scenarios against given performance indicators, as explained in the previous paragraph, systemic digital twins can also be used to solve optimization problems in a more rigorous manner. Specifically, one can define an optimization objective, typically, achieving a set of business goals at a specific point in time, and then use the systemic digital twin in reverse to address the inverse problem: determining the values of the free parameters required to meet the business goals at the specified time. Such an inverse problem can be solved using classical dynamic programming methods, with the systemic digital twin serving as a "black box" that provides the behavior of the underlying real system. The critical observation here is that, through our industrial collaborations, we found that most practical industrial optimization problems are framed in precisely this "inverse problem" way.

2. The system specification language Σ

2.1 Why creating a system specification formal language?

In order to understand better the motivation of the creation of a specific system specification formal language, let us first recall that any language can always be analyzed from three *syntactic, semantic and pragmatic* perspectives. Syntax refers to the nature of the elementary symbols managed by the language, when semantics refers to the meaning of sequences of such symbols and pragmatics to the possible practical usages of these symbols in a given operational context.

One can now use this analysis grid to compare the three main types of system specification languages that can be found in practice (see Table 1), that is to say:

1. *Unformal languages*, such as natural languages, which are used in practice by most of engineers to specify a system, with unfortunately a poor level of rigor and no real semantics, leading to many possible meanings & interpretations of a given system specification and to the impossibility of simulating such unformal system specifications;
2. *Pseudo-formal languages*, based on graphical descriptions and meta-models, such as BPMN, SysML or UML, and now widely used by systems engineers within the model-based systems engineering (MBSE) approach: they have a better level of rigor, but which is still weak due to the absence of formal semantics that leads to structural interoperability & simulation issues;
3. *Formal languages*, such as any programming language [31] or AltaRica [31], based on sound mathematical fundamentals which equips them with a formal semantics, but unfortunately not very used by systems engineers, at the exception of the safety domain: they especially support simulation due to their high level of rigor which suppresses specification ambiguity.

Our grid of analysis can indeed be typically applied to these three kinds of languages:

- *Natural languages*: consider for instance the Greek language, where a syntax could consist in the sequence ζωα of three Greek letters, whose semantics is “animals” and which is a word that can be pragmatically used in any conversation about animals,
- P*seudo-formal languages:* consider for example here the model-based systems engineering graphical description language SysML, where a syntax could be formed by an oval containing the letters “Operations”, whose semantics is the operational lifecycle phase of a system and that can for instance be pragmatically used to do lifecycle-oriented system design,
- *Formal languages*, i.e., languages with a sound mathematical semantics: let us consider here the example of a programming language such as Pascal, where a syntax could be formed of the sequence “X := X+1” of symbols, whose semantics is provided by the predicate, in the meaning of mathematical logic, “X = X0 → X = X0+1” which expresses the fact that the variable X will have the value X0+1 after processing of the considered sequence of symbols, if it initially had the value X0, which can be pragmatically used by developing for instance a user-oriented software application.

This being recalled, the key point to stress here is that only formal languages can really be simulated without ambiguity, since being able to simulate a specification language means that there is only one – and only one – interpretation for each part of a given specification. As a consequence, the Σ modeling language, which is a formal language fully dedicated to industrial system specification on which relies our systemic digital twin approach, naturally supports simulation of the systems that it models.

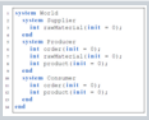
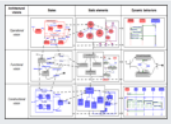
Modeling language type	Syntax used by the modeling language	Examples	Features	Fundamentals	Level of rigor	Simulation capability
Formal	Formal specification language		Formal semantics leading to compiling, simulation & strong interoperability	Mathematics	Strong	Possible
Pseudo-formal	Graphical language		No formal semantics leading to structural interoperability & simulation issues	Meta-model	Weak	Difficult since it requires a simulation semantics
Unformal	Natural language		No semantics at all leading to many possible meanings	Practice	Poor	Impossible

Table 1 – Comparing the three main types of specification languages

This is the key property that motivated the introduction of a new system specification formal language such as Σ. Note finally that its scope of application is specifically dedicated to industrial systems within their manufacturing, operation and maintenance phases where the system behavior can be typically described through a *discrete-event* approach.

2.2 Σ follows the S2ML+X paradigm

Let us first recall that any system model has always to describe both the structure and the behavior of a system. The S2ML+X paradigm, introduced by Batteux et alii in [16], consists then in claiming that a system modeling (formal) language shall consist in the combination of a mathematical framework dedicated to the description of the behavior of the system of interest and of a structuring paradigm, reflecting the generic principles of organization of any system, which is used to organize the model. This paradigm was abstracted from the safety-oriented language AltaRica developed by Rauzy [17] in the last decades and can be synthesized by the "equation": *System Models = Structures + Behaviors*.

This approach considers that structures are largely independent of any concrete system: they can thus be described through a generic systems structure description language *S2ML* [16-18]. On the other hand, behaviors can be mathematically modeled in many different ways (e.g. through continuous or discrete modeling), the choice of the relevant *mathematical framework (X)* depending on the nature of the system of interest. The previous "equation" transforms then in terms of specification languages into the new "equation" stating that any system specification language can be expressed according to a S2ML + X paradigm, which is also the case for the system specification language Σ that we used to support automatic generation of systemic digital twins of industrial enterprises.

In this last matter, we shall first introduce the key generic constructions of the *Systems Structure Modeling Language (S2ML)* invented by Rauzy (see [16], i.e., port, connection, container, composition, aggregation, prototyping/cloning, class/instantiation and inheritance, as described in Figure 6. We refer to [27], for all details on the S2ML structural modeling language.

For our system specification formal language Σ, which is at the very core of our systemic digital twin approach, the "X" of the Sys2ML + X paradigm is then the functional framework for a system, proposed by Krob in [28], applied only with a discrete time scale.

To this aim, let us first recall that a time scale T is a totally ordered set with a unique minimal element – usually denoted t_0 – and where each element $t \in T$ has a (unique) least upper bound within the time scale, called its successor and denoted t+ within T. Up to rescaling, two types of time scales are key in

practice: discrete time scales where t+ = t + 1 and continuous time scales where t+ = t + dt where dt stands for an infinitesimal quantity. Discrete time scales model event-oriented systems (such as software or manufacturing systems which are regulated by a discrete clock) when continuous time scales are used to model physical continuous phenomena.

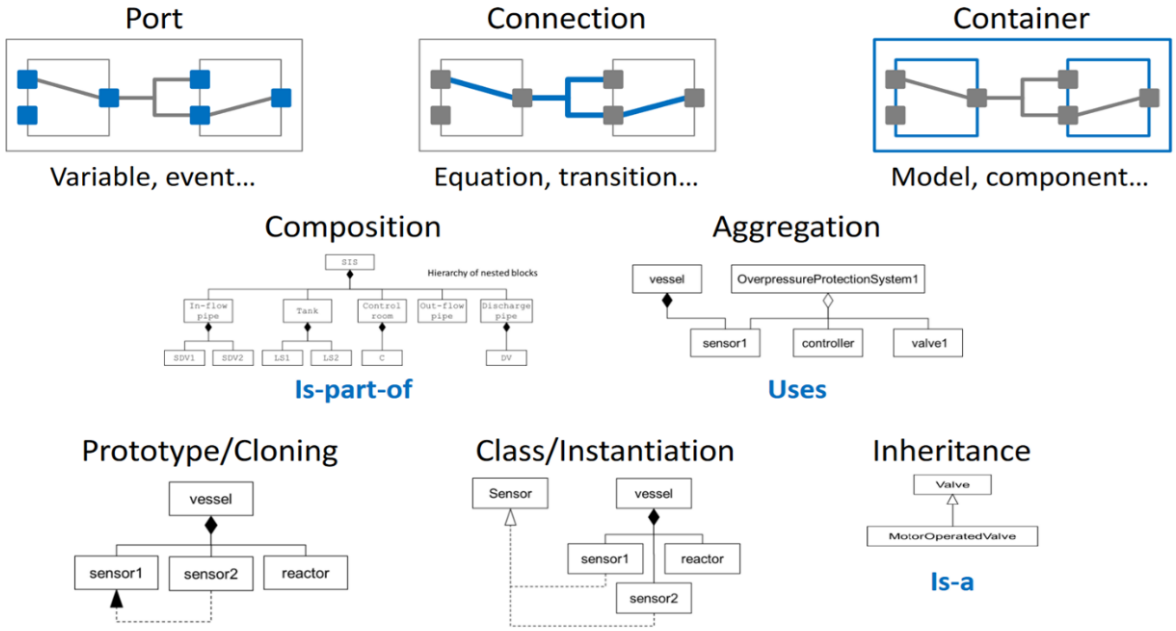


Figure 6 – The key generic constructions of the Systems Structure Modeling Language (S2ML)

One can now formally define a *system* in this framework, on one hand, by an input set X, an output set Y and a set of internal variables Q, and, on the other hand, by the following two kinds of behavioral functions that link these systemic variables among a given time scale T:

- an *external behavioral function* that produces an output y(t+) ∈ Y at each moment of time t ∈ T, depending on the current input x(t) ∈ X and internal state q(t) ∈ Q of the system ;
- an *internal behavioral function* that results in the evolution q(t+) ∈ Q of the internal state of the system at each moment of time t ∈ T, under the action of an input x(t) ∈ X on the system.

This last definition underlies the classical symbolic representation of a system as depicted in Figure 7 that we already used among the structural constructions elicited in Figure 6.

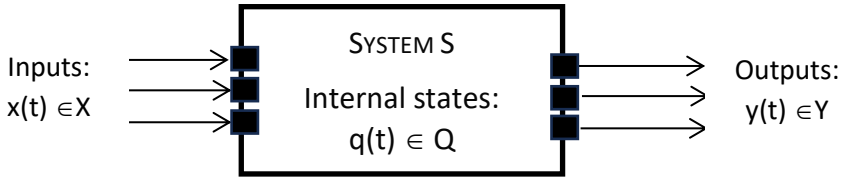


Figure 7 – Symbolic representation of a system

2.3 The core features of Σ

We are now in position to describe the *Σ modeling language* which allows first naturally to specify the hierarchical structure and the behaviors, i.e., the business processes, of a given industrial system.

On one hand, *system structures* are specified in Σ™ in a quite intuitive way. The fact that a system can be a part of another system within an hierarchy of systems, is captured through the keyword **system** in Σ, and the associated **system system …. end end** construction, which expresses that a system can contain another system, in the Σ meaning, as illustrated in the below example:

```
system World
  system Supplier
    int rawMaterial (init = 0);
  end
  system Producer
    int order (init = 0);
    int rawMaterial (init = 0);
    int product (init = 0);
  end
  system Consumer
    int order (init = 0);
    int product (init = 0);
  end
end
```

Note that the variables associated with a system are value holders that are used to describe the state of the system or to model stocks or flows managed by a system.

On the other hand, *system behaviors* are specified in Σ through the keyword **activity** which models a functional behavior, or equivalently a business process in our enterprise modeling context, associated with a given system. To specify an activity associated with a given system, one needs to explicitly define the logical condition that triggers a given activity, what shall be done when the activity starts and when the activity stops and what is the duration – in a certain unit of time that has to be fixed by the modeling engineer and used all along the Σ model – of the activity, using respectively to these different purposes the keywords "**trigger**", "**start**", "**completion**" and "**duration**", as illustrated on the next example that continues the previous one. We specified here the activity that renews the stock of raw material of the supplier of our world by asking that it begins only when the stock of raw material is less that 1,000 units and that one is not already renewing it, by stating that one is renewing at the start of the activity and that the stock of raw material was increased on 100 units, while renewing is finished, at the end of activity, and finally by defining the duration of the activity which is 30 units of time:

```
system World.Supplier
  int rawMaterial (init = 0);
  bool renewing (init = false);
end

activity World.Supplier.RenewRawMaterialStock
  trigger:
    return rawMaterial <= 1000 and not renewing ;
  start:
    renewing = true ;
  completion: {
    renewing = false ;
    rawMaterial += 100 ;
  }
  duration:
    return 30 ;
end
```

Note also the classical object-oriented mechanism that allows to specify in Σ the positioning of a given system in the system hierarchy and the allocation of an activity to a given system.

A key specificity of Σ is then that *stochastic behaviors* can be captured in Σ. This can be done in two different ways, either via random variables manipulated by activities or via random activity durations. In this matter, one can express in Σ such stochastic behaviors, either through exact probabilistic laws, such as Normal laws, uniform laws, exponential laws, etc., or through empirical distributions, i.e., experimental timed sequences. These mechanisms are illustrated in the below example, continuing the previous ones, where the amount of products consumed by the ConsumeProduct activity of the consumer follows an uniform law between 1 and 2.5 products, when the duration of the consumption follows a normal law with 10 units of time as mean and 1.5 units of time as standard deviation:

```
activity World.Consumer.ConsumeProduct
  trigger:
    return product >= 1000 and not consuming ;
  start:
    consuming = true ;
  completion: {
    consuming = false ;
    product -= uniformDeviate(1,2.5);
  }
  duration:
    return normalDeviate(10,1.5) ;
end
```

Stochastic simulations can be especially monitored within Σ in two different ways, either through *observers* which are updated continuously during the execution of a Σ model, or through *indicators* that are computed from observers, at certain moments of time, typically when the execution of the simulation of a given model is complete, as illustrated in the below examples that continues the previous examples, where we specified two observers, the first one being a numerical observer that provides the level of raw material produced by the producer and the second one being a Boolean observer that tells us whether this last level of raw material is under 30 units of raw material:

```
system World.Producer
  …
  int rawMaterial (init = 0);
  …
  observer levelRawMaterial = rawMaterial ;
  …
  observer lowLevelRawMaterial = rawMaterial <= 30 ;
  …
end
```

In order to manage *systemic scenarios*, one can also especially define *parameters* in Σ. Each parameter corresponds indeed to a value that can be modified by the end-user: defining a systemic scenario consists then just in defining the values of a given set of parameters, which can be done through a specific generic user interface, automatically created by the WorldLab platform when a given Σ model is compiled. This parametrization mechanism is illustrated in the below example, continuing again the previous examples, where the renewRawMaterialStock activity of the supplier is now parametrized with three parameters, all initialized with an initial value corresponding to the one used in our initial example, corresponding respectively to the renewal threshold, amount and duration:

```
system World.Supplier
  parameter int renewalThreshold = 1000 ;
  parameter int renewalAmount = 100 ;
  parameter int renewalDuration = 30 ;
  int rawMaterial (init = 0);
  bool renewing (init = false);
end

activity World.Supplier.renewRawMaterialStock
   trigger:
     return rawMaterial <= renewalThreshold and not renewing ;
   start:
    renewing = true ;
  completion: {
    renewing = false ;
    rawMaterial += renewalAmount ;
  }
  duration:
    return renewalDuration ;
end
```

Last, but not least, Σ also allows to *manage deformable systems*, that is to say systems whose structure or behaviors change among time (see [29] for more details). One may also point out that Σ is directly connected to a XML-like language, called WIDL, for WorldLab Interface Description Language, which allows to specify the end-user interface of a systemic digital twin and to connect the business indicators & alerts defined in a Σ model that are to be shown to the business users during the use of a systemic digital twin generated with the WorldLab platform (see [30] for more details).

2.4 Comparison with other digital twin design & development approaches

Due to its unicity, the Σ-based approach for developing a systemic digital twin can be easily compared to the other existing methods for designing a digital twin.

In this matter, let us first recall that the existing digital twin frameworks [1, 4, 7, 13] do not provide practical detailed step-by-step methods for concretely developing a given digital twin for industrial operations, but only general development guidelines. The same comment applies for the modeling frameworks that were proposed along time to support the development of a digital twin [32,33,34,35]. In the same way, one may also note that the ISO/IEC 30173 digital twin standard [36] remains quite generic and rather dedicated to a clarification of the core digital twin concepts & terminology.

The same analysis works moreover also for the more specialized literature dedicated to the topics of evolutive maintenance and validation of an industrial digital twin [37,38,39,40]. This literature indeed deals mainly again with general good practices on these issues, but not with practical step-by-step specific methods, like the one we propose in this paper.

The approaches that are the closest to our own formal approach, since they are proposing methods for automatic generation of digital twins, can be found in [41,42]. They are however quite rare among the digital twin literature and very different from our Σ-based approach, due to their radically different mathematical fundamentals. We must indeed recall that Σ is based on mathematical semantics and logics, as explained in the previous subsection 2.1, which is not the case for these other approaches which are just using classical graph theory as mathematical fundamentals.

The analysis of the existing digital twin design & development literature shows therefore that our Σ-based approach is still quite unique.

3. Developing a systemic digital twin with Σ in practice

We shall now show how to design, construct and validate in practice a systemic digital twin with WorldLab and Σ, through a case study that was constructed by abstracting and simplifying an actual industrial case on which we worked. We indeed wanted here to be both realistic and not too complex, in order to remain simple, without being simplistic. In this matter, let us therefore introduce the *port case study* that we shall follow in this section in order to illustrate our approach**.**

Due to environmental regulations, our port has to replace an old coal traffic by a new container traffic (see Figure 8), which has a huge impact on the port infrastructures since coal and containers require totally different logistics, namely train and truck logistics. One shall therefore identify and secure the investments that have to be done by the port in order to adapt it to the new traffic and to manage the decreasing of the old traffic.



Coal traffic *Container traffic*

Figure 8 – The transformation of the considered port

These elements of context being recalled, let us now see how an associated systemic digital twin can be developed according to a standard methodology, initially skteched in [31], whose main elementary steps within its three design, construction and validation phases are illustrated in the below Figure 9, in order to support the port industrial transformation.

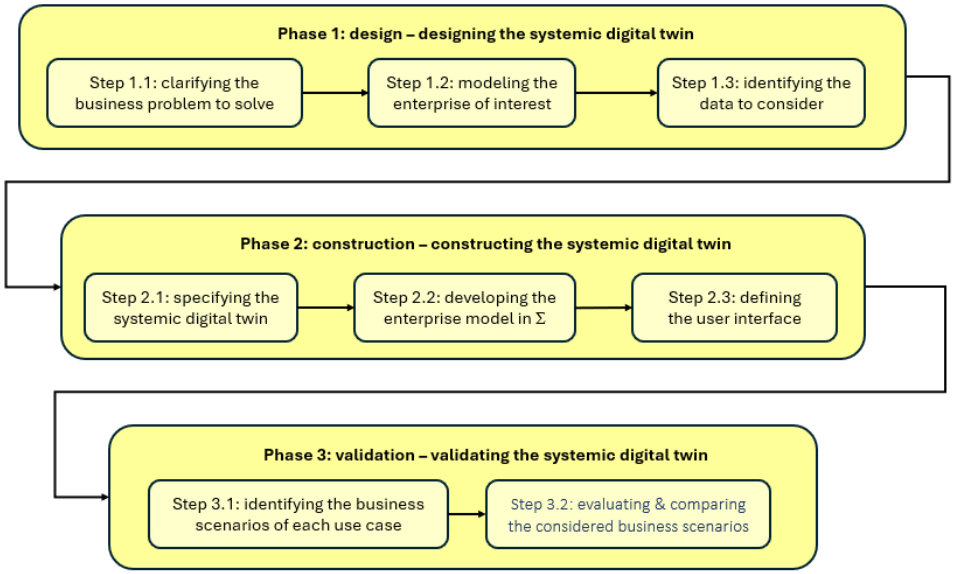


Figure 9 – Our standard methodology for developing a systemic digital twin

3.1 Phase 1: design – designing the systemic digital twin

The very first phase towards a systemic digital twin for the considered port is the design phase. Its purpose is to scope the future systemic digital twin by constructing a first informal model of the port of interest, which shall be challenged & validated before going to the next Σ modeling phase. The key point to have in mind is that one cannot go to the formal modeling phase before having stabilized our understanding of the port's transformation. Since the resulting system model will fundamentally be unstable during this phase, we are therefore using here an unformal system modeling approach which is well adapted to this situation.

In order to move towards a systemic digital twin for an industrial system, this very first phase is an initiation phase that intends understanding the relevant scope of interest. It concretely consists in fully understanding the problem(s) to be solved and the relevant scopes from strategic, business & technical and data perspectives. One shall indeed never construct a systemic digital twin for the pleasure of constructing a systemic digital twin: it shall always answer to a specific business problem whose business specificities shall normally highly guide the design of the systemic digital twin.

3.1.1. Step 1.1: clarifying the business problem to solve

In order to construct a systemic digital twin, the very first activity consists now in eliciting the business strategy which supports its construction by finding the answers to the following questions:

- What are the business objectives to achieve?
- What are the strategic or operational decisions that one wants to make and in which context?
- What are the key performance indicators (KPIs) that one wants to improve/monitor?

The above table elicits for instance the business strategy of the port that we are considering in our case study with respect to its coal-to-container transformation.

Key questions to answer	**Business strategy of the port**
What are the business objectives to achieve?	The main business objective of the port is to secure its investments, as required from switching from a coal-dominant to a container-dominant traffic, by verifying that the "to-be" port infrastructures will be resilient against the strong increase in container volumes which is expected from now to 2035.
What are the strategic or operational decisions that one wants to make and in which context?	Since the container traffic shall be managed by trucks that shall leave the port either by road, or by train, the key decisions that the port has to take are of two kinds: 1) How to transform & optimize the existing infrastructures of the container terminal in order to face the expected increasing of the container traffic? 2) How to size a new rail-road terminal that shall manage a part of the traffic induced by the future development of the container traffic?
What are the key performance indicators (KPIs) that one wants to improve/monitor?	• KPI #1: amount of new port infrastructures (cranes, gates, reach stackers) which are required by the transformation of the port • KPI #2: customer quality of service (queuing time, volume of managed containers) • KPI #3: level of pollution induced by the new port infrastructures

Table 2 – Elicitation of the business strategy of the port of our case study

3.1.2 Step 1.2: modeling the enterprise of interest

The second activity of the initiation phase of the construction of a systemic digital twin consists in modeling the industrial system of interest, which can be achieved through a series of inter-related *modeling steps*, consisting in defining respectively the system breakdown, the functional interactions managed by the system of interest and finally its use cases, i.e., in eliciting its structural, functional and operational architecture, according to the CESAM methodology [23], applied however here in a reverse way since we are not designing a future system, but analyzing an existing one.

In the context of the considered port transformation, the first modeling step is a structural modeling step consisting in a system breakdown analysis that leads to the system breakdown structure which is illustrated in Figure 10. One can see there the complete structural breakdown of the perimeter of interest – called here "World" – that synthesis all the systems involved/impacted by the transformation of the port that we considered. This system breakdown splits in two parts:

- The port which can broken down in four subsystems:
  - The internal road infrastructure of the port used by the trucks that are transporting by road the containers managed by the port,
  - A container terminal where the containers managed by the port are coming & leaving by sea through a specific cargo logistic, stored in a container storage park and coming & leaving by road through trucks that enter in the container terminal by the terminal access,
  - A number of existing & forecasted new warehouses, where trucks are bringing & taking containers in order their goods to be stored there,
  - A new expected rail-road terminal consisting in train & truck logistic facilities and in a stock of reach-stackers (specific machines used for moving containers between trains and trucks).
- Three external systems, that is to say:
  - The natural environment that will be especially impacted by the pollution produced by the increasing of the truck traffic induced by the growth of the container traffic,
  - The sea from / to where are coming / leaving the ships that transport the containers that are managed by our port,
  - Two external road & rail infrastructures, respectively used by the trucks and railways that are involved in the management of the container traffic of the port.

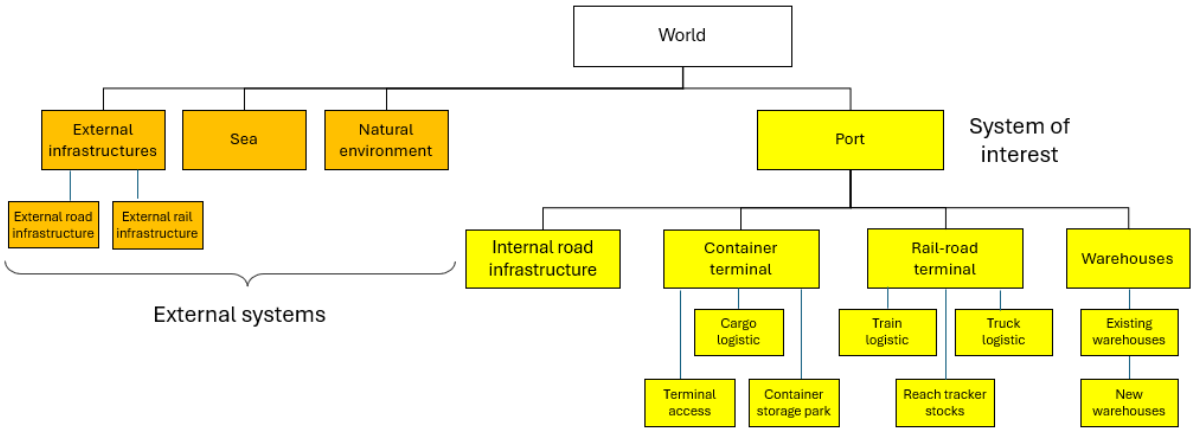


Figure 10 – System breakdown structure of our port industrial environment

Knowing the system breakdown structure of the system of interest, we can move to the next functional modeling step that deals with a *functional interactions analysis***,** aiming at identifying the structuring *flows and interactions* that do exist within the various systems forming the environment of the system of interest, and the core *activities* performed by these systems. It results in a *functional interaction diagram*, that synthesizes all the interactions existing between the key subsystems of the system of interest and its external systems, and also highlights the key activities achieved by these subsystems.

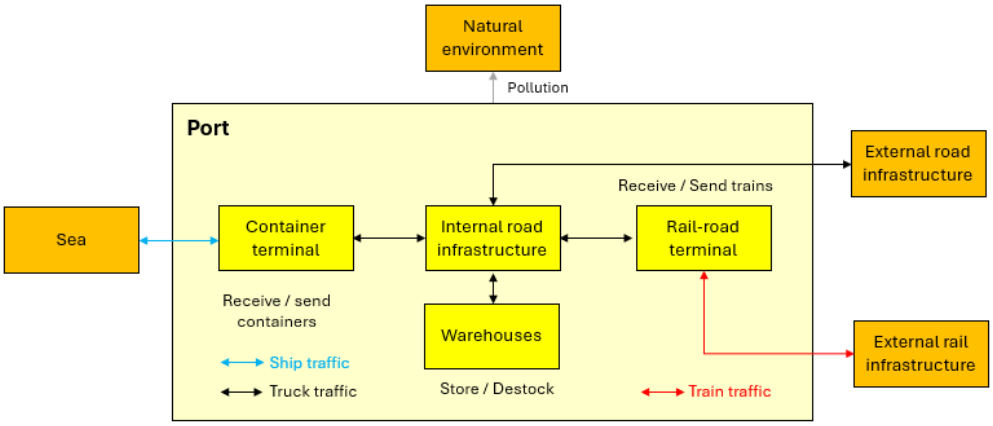


Figure 11 – Functional interactions involved in the considered port industrial environment

In the context of our port transformation, this new functional interactions analysis leads to the functional interaction diagram illustrated in Figure 11. This new diagram highlights the main functional chain involved in our case study, that is to say the fact that the container traffic is triggered by the sea from / to where are coming / going the ships that are transporting the containers and handled by the port and by external road & railway infrastructures. It especially shows that the port internal container two-way traffic is managed through a container terminal, connected to the sea and to the ship traffic, where containers are therefore received and sent, and an internal road infrastructure for trucks, which connects the container terminal to the external road infrastructure, but also to a rail-road terminal, itself connected to the external rail infrastructure, where trains are received and sent, and to internal warehouses that can be seen as temporary buffers where the goods contained in the containers are stored and destocked among time. Note finally that we also put there a specific stress on the pollution flow that goes from the port to the natural environment.

The last system modeling step is an operational modeling step that consists in formalizing the business strategy in terms of *use cases* of the system of interest that shall be modeled, simulated & analyzed with a systemic digital twin in a next phase.

In the context of our port transformation case study, these use cases are for instance the following:

- *Use case 1 – Sizing of the unloading cranes fleet within the container terminal* – The increasing of the container traffic will require to add new unloading cranes to the port, which is a very expensive investment that has to be finely planed. How much new cranes are therefore required and when shall they be put in service?
- *Use case 2 – Optimization of the truck access to the container terminal* – The increasing of the container traffic will increase the truck traffic on the port. How shall one thus reorganize the truck traffic management in order to optimize the in/out access of trucks to the container terminal and to provide an optimal quality of service to the trucks?

- *Use case 3 – Optimization of rail-road terminal infrastructure* – The increasing of the container traffic will require a new rail-road terminal within the port. How shall one therefore organize optimally this new terminal at the interface between trucks and railways in terms of specific loading/unloading machines?

As a synthesis, we shall therefore remind that the key deliverables of the modelling activity are:

- Structural modeling: the *system breakdown structure* of the perimeter of interest,
- Functional modeling: the *functional interactions diagram* of the system of interest,
- Operational modeling: the *use cases* to analyse.

3.1.3 Step 1.3: identifying the data to consider

To identify the *data* that one needs to consider for constructing a systemic digital twin of a given system of interest, one needs to go back to the functional interactions analysis that provides the main flows to take into account, each flow reflecting in a corresponding set of data.

In our port case study, all these flows are especially triggered by the import / export container flow which is associated with the ship traffic. In other words, each container managed by the port, either comes initially from a ship or ends finally on a ship, meanwhile being, either transported by a truck or a train, or stored temporarily in a warehouse. According to this analysis, two main types of data, respectively associated with the pushed imported container flow coming from the sea and the pulled exported container flow going to the sea, are required to achieve the core of a systemic digital twin for the port that we are considering, that is to say:

1) the *existing & expected volumes of imported & exported container flows* managed by the container terminal,
2) the *distribution among time of these volumes* within the various systems involved in the perimeter of interest.

The below Figure 12 illustrates the corresponding data that have to be identified.

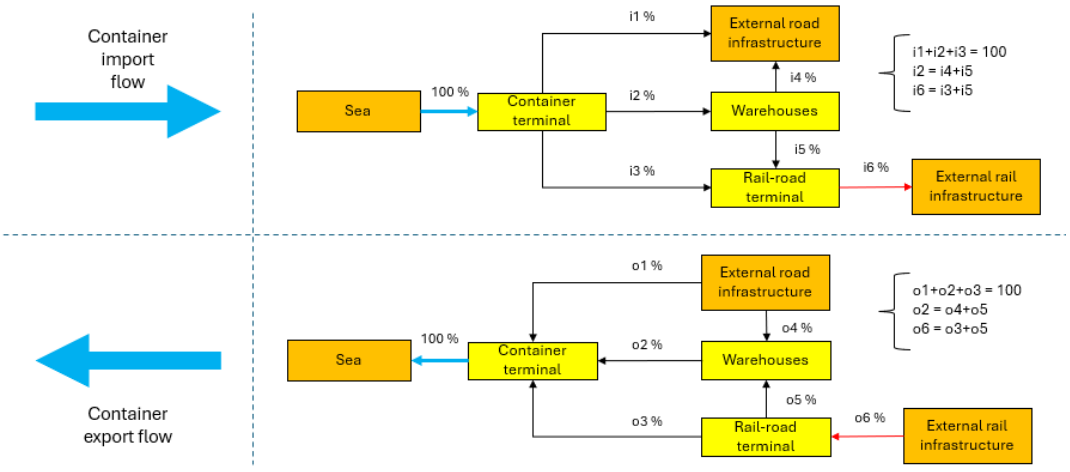


Figure 12 – The key data associated with the container flows managed by our port

Additional data is also required to estimate the level of pollution produced by the container traffic managed by the port of our case study. Note that this data can typically be obtained by understanding the proportionality relationship existing between the volume of containers managed by the port among time and such an observable which may be of the data of our model.

The below table synthesizes then all the data required to construct a systemic digital twin of our system of interest for our port case study.

Flow	Data to consider
Container import flow	• Volumes of sea-managed imported containers per unit of time • Relative volumes managed within the container import network
Container export flows	• Volumes of sea-managed exported containers per unit of time • Relative volumes managed within the container export network
Pollution	• Volume of pollution produced per managed container per year

Table 3 – Data to consider in order to construct a systemic digital twin for the considered port

3.2 Phase 2: construction – constructing the systemic digital twin

In order to construct a systemic digital twin for an industrial system, the second phase consists in first specifying finely the systemic digital twin, based on the material coming from the design phase, which starts by defining the simulation model architecture and ends by achieving a data analysis in order to understand the specific data that will be used by the simulation model, before developing – using the WorldLab™ platform – the concrete systemic digital twin of the system of interest, based on the material coming from the previous specification step which consists in implementing the core Σ model on which relies the systemic digital twin and its user interface through a WIDL specification.

The key point to understand in this matter is the fact that the simulation model, that shall be ultimately executed, on which relies a systemic digital twin, is of course derived from the system model developed in the design phase, but also generally quite different. One indeed needs to take here into account both computing constraints in order to get efficient simulations and specificities of the Σ language that are constraining, but also possibly simplifying, the simulation model.

3.2.1 Step 2.1: specifying the systemic digital twin

The very first activity of the specification step consists in *defining the architecture of the simulation model* of the system of interest – to be implemented in Σ – that shall be at the core of its targeted systemic digital twin, as ultimately generated by WorldLab.

More specifically, one has to define here, based on the inputs of the design phase, the structure, the Σ activities (with their sequencing) and the observers of the perimeter of interest, as they shall be implemented, since implementation brings its own constraints and is usually not a copy/paste of the initial system analysis, as already mentioned above. In this matter, the relations between the inputs of the design phase and their counterparts in terms of simulation model architecture are the following:

- The system breakdown of the system of interest shall first reflect in the systemic structure of perimeter of interest, using the hierarchical system concept of Σ,
- The main activities & interactions within the perimeter of interest provided by its functional interaction diagram reflect then in the precise definition & sequencing of the Σ activities,
- Finally, the key performance indicators associated with the business objectives to fulfil shall reflect in the definition of observers in the Σ meaning.

The crucial part of the specification step consists therefore in identifying the activities – in the Σ meaning – and the key resources that they are managing / consuming, as associated with the different

systems of the perimeter of interest, together with their sequencing within the simulation model. In this last matter, one shall especially remember that the simulation engine will repeat these activities for each chosen time step, up to the scheduled end of a simulation. Note also that one already has to arbitrate here between introducing an activity or using an observer, to model each part of the initial system model, as defined during the design phase.

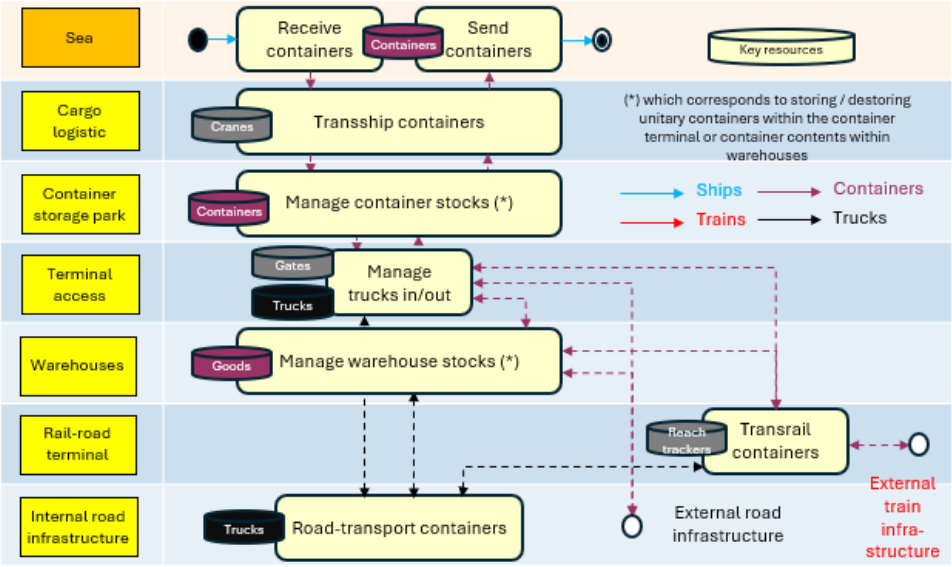


Figure 13 – Σ activities of the simulation model that describes the behaviour of our port case

In our port case, the Σ activities deal first with the containers, by receiving / sending containers from / to the sea, transhipping them through the cargo logistics, managing their stocks in the container storage park and managing the stocks of goods contained in the containers at the level of the warehouses, secondly with the trucks that are transporting containers by managing their in / out access in the container terminal access and their road transportation within the port and finally with railways where one needs to transrail containers. Concerning trucks, we shall suppose that they are available when required, which allows to abstract them, excepted at the levels of the two "manage truck in/out" and "road-transport containers" activities. We synthesized these activities in the BPMN-like diagram of Figure 13 where we also highlighted all the resources (containers, cranes, gates, trucks, goods, reach trackers) that are supporting these different Σ activities.

To define then the relevant *observers* to implement, a natural way is to associate one specific observer with each business Key Performance Indicator (KPI), as initially identified in the design phase, and to complete these first mandatory observers by additional ones, associated with complementary information of interest that one may possibly also need to provide.

In our port case, the key performance indicators to manage are the following ones:

- KPI #1: amount of new infrastructures (cranes, gates, reach trackers) required by the port
- KPI #2: customer quality of service (queuing time, volume of managed containers)
- KPI #3: level of pollution induced by the new infrastructures

As a consequence, three types of observers – in the Σ meaning – shall therefore be defined to measure these key performance indicators.

The second activity of the specification phase consists then in *analysing the business data* – which requires their preliminary capture, that appears in practice to be always a rather complicated activity

– in order to identify the probabilistic laws, if any, that govern them and the main dimensioning relationships that are key for implementing a simulation model.

This data analysis activity is highly specific to each application case. In our port case, it consists in analysing the import / export past & future container volumes, capturing the relative containers volumes that are managed by its logistic network and identifying the dimensioning factors associated with the mandatory observers. The corresponding data analysis is elicited here below.

The analysis of imported container volume data in our port showed first that all monthly imported traffics of a given year are distributed according to Normal laws**,** whose means & standard deviations evolve among time in the same way, i.e. proportionally to the port traffic growth, since the observed ratio between these means & standard deviations remains rather stable.

In this last matter, we shall remember that a Normal law $\mathcal{N}(\mu; \sigma)$ of mean μ and standard deviation σ is a random variable N(μ; σ), which is characterized by the following probability law:

$$P(\, a \le N(\mu;\sigma) \le b\,) = \frac{1}{\sqrt{2\pi}\,\sigma} \int_a^b \frac{\exp\left(-(x-\mu)^2\right)}{\sigma^2}\, dx\ ,$$

whose density has a classical bell-shape. The distribution of monthly container volumes in a given year follows therefore a Normal law of mean μ and standard deviation σ if the following approximation relation is statistically valid:

$$\frac{\text{Cumulated amount of containers from january to month m within year y}}{\text{Total amount of containers for year y}} \approx P\,(\,1 \le N(\mu;\sigma)\ \le m\,)\ .$$

This last property can be observed on Figure 14 which gives the historical imported container monthly volumes for the period 2016 – 2021, as provided by the port management, with indication of the associated means, standard deviations and mean to standard deviation ratios. Data analysis especially showed that the different distributions of imported container volumes do follow a Normal law for each year involved in our set of data, as confirmed by a χ^2 test. An example of the fit between the actual data and their Normal law modelling is shown on Figure 14 for the 2016 imported container volumes, which can be captured by a Normal law of mean 12,000 and standard deviation 1,500.

Import	2016	2017	2018	2019	2020	2021
January	12 500	13 750	13 750	16 000	14 000	24 750
February	15 500	12 250	13 250	15 000	16 000	22 250
March	11 500	10 750	14 750	17 500	18 500	15 250
April	11 500	16 250	16 750	17 500	16 000	20 250
May	12 000	13 750	17 250	17 000	18 000	20 250
June	14 000	14 250	15 750	15 500	13 000	20 250
July	12 000	11 250	13 250	19 000	21 000	21 750
August	12 250	15 750	14 750	18 000	14 000	28 250
September	10 500	13 750	13 750	13 000	18 500	18 750
October	12 000	13 750	19 250	17 500	19 000	20 250
November	9 500	12 250	12 750	12 500	20 000	22 250
December	12 000	10 750	12 250	16 500	22 500	24 250
Total	145 250	158 500	177 500	195 000	210 500	258 500
Growth	N/A	9%	12%	10%	8%	23%
Mean	12 000	13 000	15 000	16 500	17 500	21 500
Standard deviation	1500	2000	2000	2000	3000	3500
Mean/ standard deviation	8	7	8	8	6	6

Covid 2019 crisis im

(Ctrl)

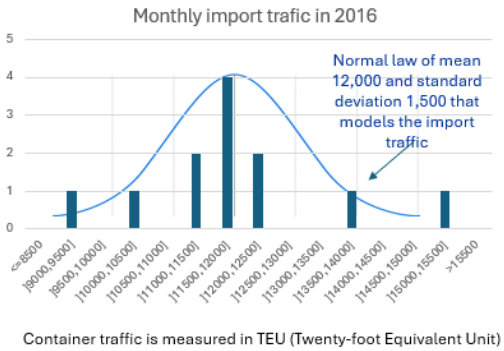


Figure 14 – Data analysis of the imported container volumes in our port case

In the same way, the analysis of the exported container volume data for our port showed that the monthly exported traffics of a given year are also distributed according to Normal laws whose means & standard deviations evolve among time in the same way, i.e. proportionally to the port traffic

growth, since observed ratios between these means & standard deviations remain again quite stable, which synthesizes our data analysis for exported container volumes.

For the future container volumes managed by the considered port, we shall indeed assume that they will have a constant growth per year up to 2035, symmetrically both for the imported & exported container traffics, starting from year 2021 baseline, that distributes per month according to the probabilistic pattern that we identified through the previous historical data analysis. Note that these values for the future traffic growth may however be considered as parameters of our systemic digital twin in order to construct various evolution scenarios for our port case study.

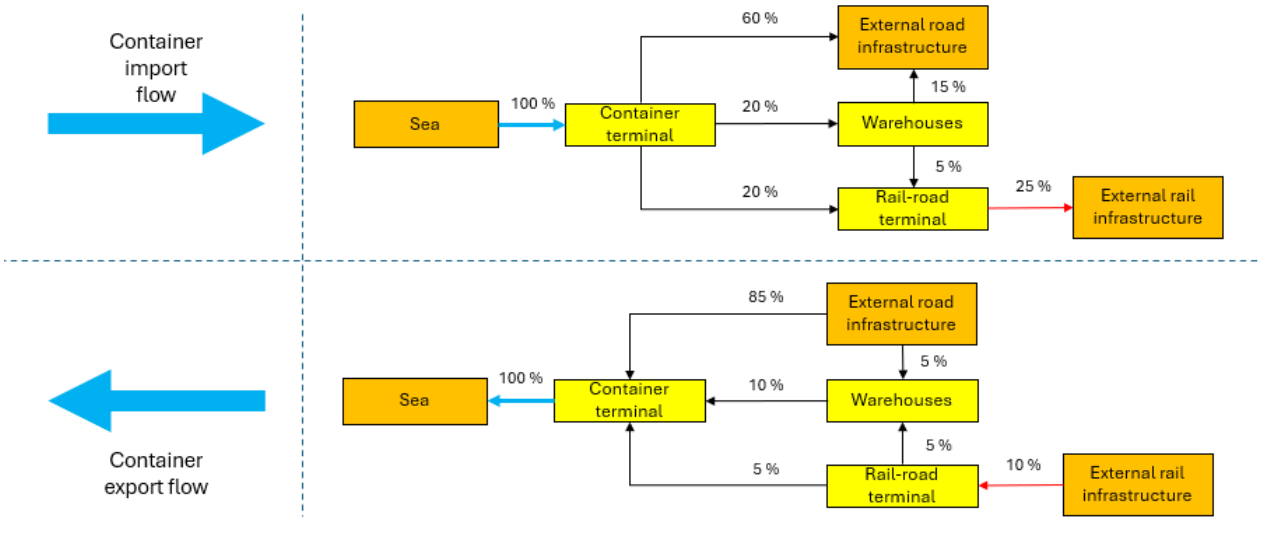


Figure 15 – Distribution of the import & export traffic within the port logistic network

The key point is that the understanding of the probabilistic behaviours in the past, based on historical data analysis, allows us indeed to model the probabilistic behaviours for the future, that shall be made according to the probabilistic Normal pattern observed in the past, taking into account the projected evolution of the future business data. Here, the baseline imported & exported container volumes are following known Normal laws in 2021, which are used to construct similar Normal laws for the imported & exported container volumes for any year between 2022 and 2035, by considering that their means and their standard deviations are obtained by increasing each year their 2021 baseline values by a multiplicative constant $1+\gamma$ reflecting the growth γ of the traffic per year, which captures both the observed Normal law pattern in the past and the fact that the observed mean to standard deviation ratios were rather stable within the historical data.

The historical relative container volumes – which were relatively stable in the past – within the logistic network of our part are provided in Figure 15. We shall suppose here that these values will not significantly evolve in the future. Note that the values for the future may however also be considered as parameters of our systemic digital twin in order to construct evolution scenarios for our port.

Finally the last step of the data analysis activity consists in identifying all dimensioning values that are used by the various observers that were defined in the design phase. In this matter, Table 4 shows for instance all numerical values, with the corresponding business explanations, that are required to precisely specify the mandatory observers associated with our port case.

KPIs		Core dimensioning equations	Other related features
KPI #1: amount of new industrial infrastructures (cranes, gates, reach trackers) required by the port	Amount of cranes required by the cargo logistic of the port	A crane can import / export a TEU in 1.5 minutes, plus an hazard following an uniform law with min and max equal respectively to 0.5 and 1 minute.	Opening time of the port is 24 hours / 24 per day on a basis of 7 days / week.
	Amount of gates for the container terminal required by port	Access time at the gate is 1.5 minutes per truck, plus an hazard following an uniform law with min and max equal respectively to 0 and 0.5 minute.	1 truck is able to manage 1.5 TEU in average.
	Amount of reach trackers required by the rail-road terminal of the port	1 reach tracker manages loading or unloading of 1 FEU with respect to a train in 2 minutes, plus an hazard following an uniform law with min and max equal resp. to 0.5 and 1 minute.	A train can handle 40 TEU A train can stop at most 4 hours in the port.
KPI #2: customer quality of service (queuing time, volume of managed containers)	Queuing time at the container terminal of the port	Access time at the gate is 1.5 minutes per truck, plus an hazard following an uniform law with min and max equal respectively to 0 and 0.5 minute.	Queuing time shall be always under 1 hour.
	Total volume of containers managed by the port	Total volume of containers managed by the port is the sum of all monthly imported & exported containers.	The container storage park has a maximal capacity of 15,000 TEU.
KPI #3: level of pollution induced by the new infrastructures		Volume of pollution is equal to 13.1 tons per year produced for 1,000 trucks managed by the port.	1 truck is able to manage 2 TEU in average.

Table 4 – Dimensioning factors associated with the mandatory observers

3.2.2 Step 2.2: developing the enterprise model in Σ

The very first activity of the *development* of the systemic digital twin of the system of interest consists in implementing the core Σ model on which relies the systemic digital twin, based on the inputs of the specification phase. Note however that this activity can possibly lead to specification changes when specification ambiguities are elicited & resolved during implementation.

According to the double structural and behavioral nature of the Σ language (see the previous section), the resulting Σ model is organized in two main parts: the first one – quite short – describes the hierarchical structure of the perimeter of interest, here called “PortInfrastructureWorld”, when the second one provides the detailed specification of the business processes – implemented through so-called activities in Σ – managed by the various systems involved in the perimeter of interest. These two parts are illustrated below. Table 5 presents first the implementation in Σ of the structure of our perimeter of interest. Note also there the observer “monthlyPollutionToNaturalEnvironment”, which just expresses the corresponding mathematical relationship, as defined at the end of the specification phase. Table 6 and Table 7 show then respectively the implementation in Σ of one transverse process, here time management, implemented through a clock that monitors, on a monthly basis, our simulations, all starting in January 2016 and ending in December 2035, and the implementation of the container importation / exportation laws, expressing that these laws are based on historical data up to December 2021 and on a projection formula, depending on the expected annual growth of the container traffic in our port, from January 2022 up to December 2035, all of these relations being based on the Normal law-based data analysis of the initial model specification step.

```
/*
 * 2. Port Infrastructure World
 * ----------------
 */

system PortInfrastructureWorld

    // Port Infrastructure World parameters
    parameter float openingTimePerDay (unit = "Hours")= 24.0;

    // Port Infrastructure World variables
    State clockState (init = STANDBY);
    float clockYear (init = 2016, unit = "Year");
    int clockMonth (init = 1, unit = "Month");
    int constMinutesPerMonth (init = openingTimePerDay * 60 * 30.4, unit = "Minutes");


    // Port Infrastructure World global ressources
    int importedContainersToExternalRoadInfrastructure (init = 0, unit = "TEU");
    int exportedContainersFromExternalRoadInfrastructure (init = 0, unit = "TEU");
    int importedContainersToExternalRailInfrastructure (init = 0, unit = "TEU");
    int exportedContainersFromExternalRailInfrastructure (init = 0, unit = "TEU");

    //Volume of pollution is 13.1 tons per year for 1,000 trucks managed by the port.
    // unit : tons per month
    observer monthlyPollutionToNaturalEnvironment = 13.1/12 *
main.DunkirkPort.InternalRoadInfrastructure.montlyTrucksTransportMission / 1000;

    // Port Infrastructure World breakdown stucture
    system Sea ... end
    system DunkirkPort
        system ContainerTerminal
            system CargoLogistic ... end
            system ContainerStoragePark ... end
            system TerminalAccess ... end
        end
         system Warehouses ... end
         system RailRoadTerminal ... end
         system InternalRoadInfrastructure ... end
    end
end
```

Table 5 – Specifying the structure of the perimeter of interest in Σ

```
activity PortInfrastructureWorld.incrementClock
    trigger:
        return clockState == STANDBY and not (clockMonth == 12 and clockYear == 2035);
    start:
        clockState = WORKING;
    completion:{
        clockState = STANDBY;
        if clockMonth < 12 then
            clockMonth += 1 ;
        else {
            clockMonth = 1;
            clockYear += 1;
        }
    }
    duration:
        return 1;
end
```

Table 6 – Specifying in Σ the clock that manages the simulation time

```
activity PortInfrastructureWorld.Sea.ImportExportContainers
    trigger :
        return importExportContainersState == STANDBY and main.clockState == WORKING;
    start:{
        importExportContainersState = WORKING;

        if main.clockYear < 2022 and main.clockMonth==1 then {
            //ImportedContainer concern
                meanImportContainerVolumes = empiricalDistribution("data/MeanImportContainerVolumes.tsv", step, main.clockYear);
                standardDeviationImportContainerVolumes = empiricalDistribution("data/SDImportContainerVolumes.tsv", step, main.clockYear);
            //ExportedContainer concern
                meanExportContainerVolumes = empiricalDistribution("data/meanExportContainerVolumes.tsv", step, main.clockYear);
                standardDeviationExportContainerVolumes = empiricalDistribution("data/SDExportContainerVolumes.tsv", step, main.clockYear);
        }
        if main.clockYear >= 2022 and main.clockMonth==1 then {
            //ImportedContainer concern
                meanImportContainerVolumes = round(meanImportContainerVolumes * (1.0 + growthVolumeImportedcontainer/100));
                standardDeviationImportContainerVolumes = round(standardDeviationImportContainerVolumes * (1.0 + growthVolumeImportedcontainer/100));
            //ExportedContainer concern
                meanExportContainerVolumes = round(meanExportContainerVolumes * (1.0 + growthVolumeExportedcontainer/100));
                standardDeviationExportContainerVolumes = round(standardDeviationExportContainerVolumes *(1.0 + growthVolumeExportedcontainer/100));
        }
        //ImportedContainer concern
            importedContainers += round(normalDeviate(meanImportContainerVolumes,standardDeviationImportContainerVolumes));
        //ExportedContainer concern
            exportedContainers += round(normalDeviate(meanExportContainerVolumes,standardDeviationExportContainerVolumes));
    }
    completion:{
        importExportContainersState = STANDBY;
    }
    duration:
        return importExportContainersDuration;
end
```

Table 7 – Specifying in Σ the container importation / exportation laws

3.2.3 Step 2.3: defining the user interface

The *design of the user interface* of our systemic digital twin is the next crucial part of the development process since it consists in defining what shall be seen by the business user when using a systemic digital twin. The user interface shall indeed reflect the business objectives & associated KPIs and be able to cover the business use cases that were defined during the initial design phase.

In our port case, the main business user interface – for managing step-by-step simulation – is shown in Figure 16. It shall allow the business user to simulate the evolution of the container traffic within the considered port among time, under various hypotheses such as an yearly container traffic growth assumption, the capacity of the container storage park measured in TEU, the number of available cranes for transhipping the containers, the number of access gates for trucks to the container storage park and the number of reach trackers in the rail-road terminal, which are parameters that the business user can set – their values can be defined through another dedicated user interface and be seen on the right-hand side of Figure 16 – in order to define a business scenario to simulate. The scenario which is simulated in Figure 16 corresponds for instance to the situation where one is using the existing infrastructures of the port as they are and one can see that they saturate in May 2026, which means that one shall invest in order to resize them much earlier in the past.

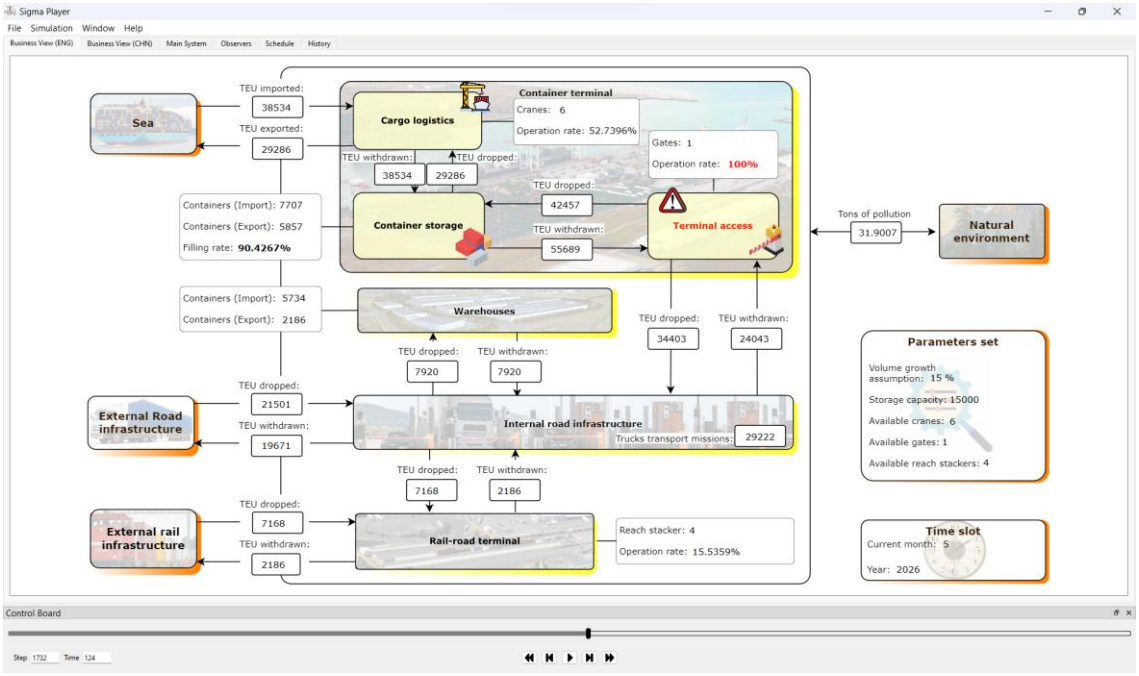


Figure 16 – The business user interface for the port case (intermediate status of a simulation)

The same simulation shall also show that the existing sizing of the container terminal of our port will ultimately lead in December 2035 to a complete saturation of all the key infrastructures (i.e., cranes, storage park, access gates) of the container terminal.

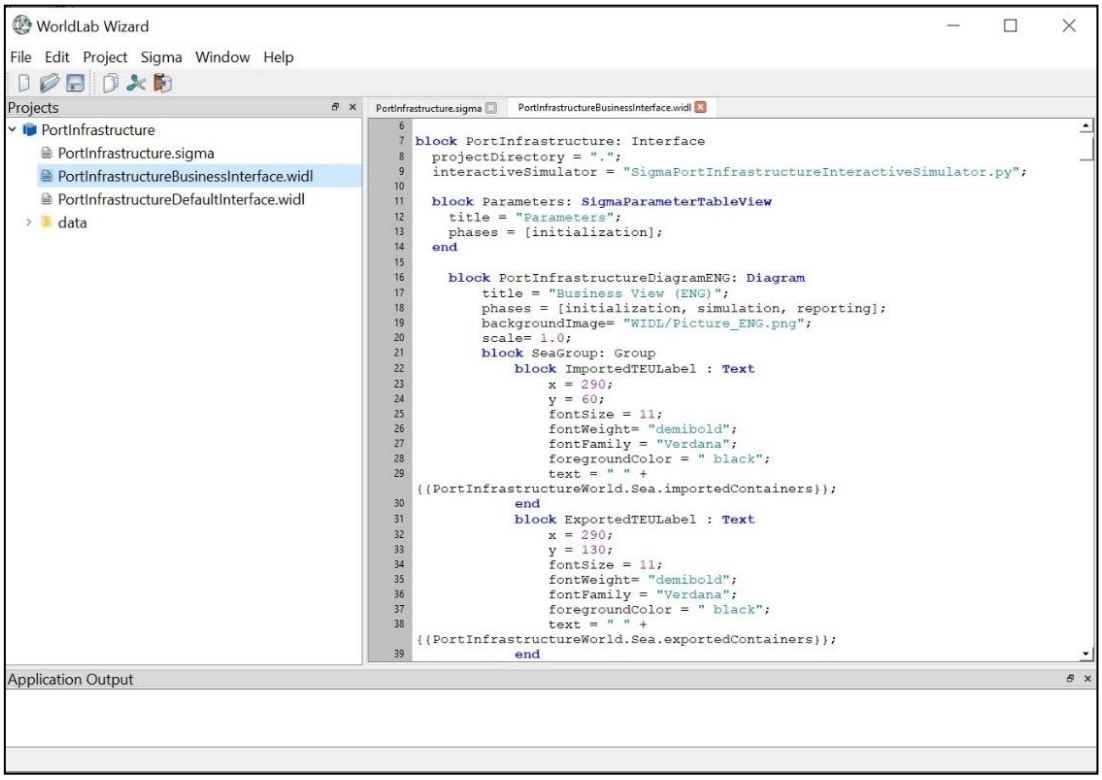


Figure 17 – The WIDL definition of the user interface of our port case

To achieve such a business interface, one shall especially develop a WIDL model that describes its structure and its connection with the underlying Σ model by specifying the exact locations where one

shall see a given value computed from the simulation of the model of the system of interest (see Figure 17 for an illustration on our port case).

3.3 Phase 3: validation – validating the systemic digital twin

The last, but not least, phase can now deal with the *validation* of the systemic digital twin of the system of interest, as actually implemented, in order to *analyze the relevant business use cases* and contribute to the business strategy as defined during the initial design phase. The typical organization of this last phase consists, for each identified use case, in identifying the business scenarios to analyze and then evaluating & comparing these business scenarios in order to find the best one. Note also finally that this validation phase can only be done by discussing & challenging with business users & experts the business relevance of the results that are provided by the systemic digital twin.

3.3.1 Step 3.1: identifying the business scenarios of each use case

The first step of the validation phase consists in identifying the *business scenarios* to analyze & evaluate, based on our systemic digital twin for the perimeter of interest, for each use case that one shall consider as provided in the initiation phase. This activity reflects in eliciting the parameters – as implemented in the Σ model of the system of interest – that define these scenarios and choosing the numerical values that are characterizing each scenario.

To illustrate this new activity, let us consider the use case 2 of our port case study, which deals with *optimization of the truck access to the container terminal*, as defined during the design phase. In this matter, we shall remember that the admission of trucks to the container terminal is a process that was carried out manually by one gate. This process was suitable for the current flow of containers which is transported by trucks on roads, but will undoubtedly pose capacity problems in the future. At this level, the business challenges that the port is facing are especially the following ones:

- Anticipate blocking of terminal access
- Control / limit the impact on air pollution of the increase in the number of trucks that are serving the container terminal.
- Avoid loss of customers due to poor quality of service
- Avoid forwarding traffic to other ports

Three progressive business scenarios can then be associated with this last use case, under a common assumption of 15 % annual growth of the container traffic within the considered port:

- Scenario 0 is the *baseline scenario* where one does nothing, that is to say where one just reuse the existing infrastructure of the port, consisting in six transshipping cranes, a container storage park with capacity of 15,000 TEU, one single gate for managing the truck access of the container terminal and a rail-road terminal with one single reach tracker (to model the current minimalist situation existing in the port);
- Scenario 1 is an *intermediate scenario* with ten transshipping cranes, a container storage park with capacity of 35,000 TEU, five gates for managing the truck access of the container terminal and a rail-road terminal with five reach trackers;
- Scenario 2 intends finally to be a *robust scenario* with thirteen transshipping cranes, a container storage park with capacity of 50,000 TEU, height gates for managing the truck access of the container terminal and a rail-road terminal with four reach trackers.

The below table synthesizes these three business scenarios that were defined in strong interaction with the port management that has the underlying business knowledge.

Parameters	System location	Scenario 0	Scenario 1	Scenario 2
Growth in import-export volumes after 2022	Sea	15 %	15 %	15 %
Number of cranes	Cargo logistics	6	19	13
Container storage park capacity	Container storage park	15,000	35,000	50,000
Number of gates for the container storage part	Terminal access	1	5	8
Number of reach-stackers	Raid-road terminal	1	5	10

Table 8 – The three scenarios associated with use case 2 that we shall consider

3.3.2 Step 3.2: evaluating & comparing the considered business scenarios

The second and last activity of the validation step consists finally in simulating and comparing the various scenarios identified during the previous activity, through stochastic simulations consisting in running a huge number, here 20,000, of stories with Monte-Carlo stochastic simulations, in order to get relevant statistical values for the key performance indicators of interest that are associated with our different business scenarios.

In the context of the analysis of the use case 2 associated with our port, the key performance indicator of interest is the monthly operation rate of the gates of the container terminal. The stochastic simulation of the baseline scenario 0 consists then for instance in playing 10,000 simulations of the Σ model of the system of interest, as achieved during the development phase, with the following data:

- One single admission gate with 90 seconds processing time at the most restrictive point,
- Each truck transports in average the equivalent of 2 TEU,
- Truck arrival follows an empirical distribution corresponding to the actual observed data,
- Future logistic flows are given by the port growth trends.

Such stochastic simulations can be easily managed with WorldLab which offers this important feature to its users for evaluating key performance indicators on a given Σ model.

In our port case, the first corresponding result is then provided in Figure 18 which gives the monthly operational use of the access gates of the container terminal – on a scale from 0 % to 100 % – for a period running from 2016 to 2035, integrating the evolution of the container traffic in the port. The line in dark blue shows here the average value of the monthly operational use of the access gates during the simulated period, when the zone in light blue corresponds to the maximum likelihood area of this indicator, defined by minus / plus one standard deviation with respect to the previous average value, and the two lines in dashed red represent the minimal and maximal values of the indicator of interest for all simulations. One can therefore immediately see that the access gates do saturate by early 2025 in this baseline scenario, where existing port infrastructures are reused without any change, and that first saturations already begin to occur during the 2023-2024 period in extremal cases.

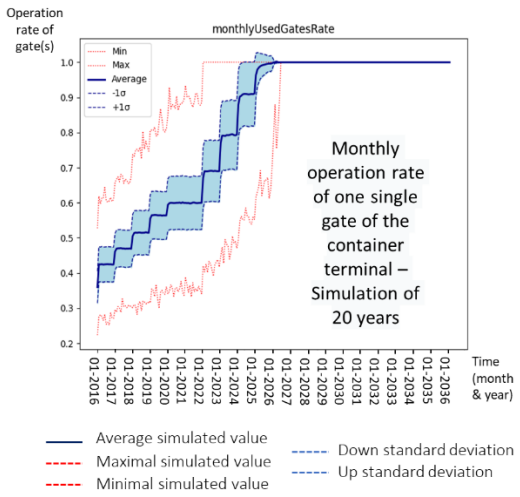


Figure 18 – Stochastic simulation of the baseline scenario 0

One can then continue to manage with WorldLab the stochastic simulations (see Figure 19) that are required in order to evaluate in the same way the two other scenarios associated with use case 2 for our port, as presented in the previous sub-section.

These new stochastic simulations especially show that:

- In scenario 1, access gate saturation occurs between 2031 to 2035 in 100 % of cases,
- In scenario 2, access gate saturation may occur between 2034 to 2035, but only for less than 15 % of cases.

To avoid gate saturation (therefore truck waiting time), the most effective solution for 2035 is thus scenario 2, consisting in adding 7 new access gates, which requires managing the construction of the corresponding resources, in order to achieve a smooth admission to the container terminal. Note also that the stochastic simulation shows that saturation may still happen in scenario 2, but only in extremal situations corresponding to 15 % of the cases: the stochastic simulation shows therefore that there is a business trade-off to do here between adding more gates – which has a cost – and accepting possible saturation in a limited number of situations.

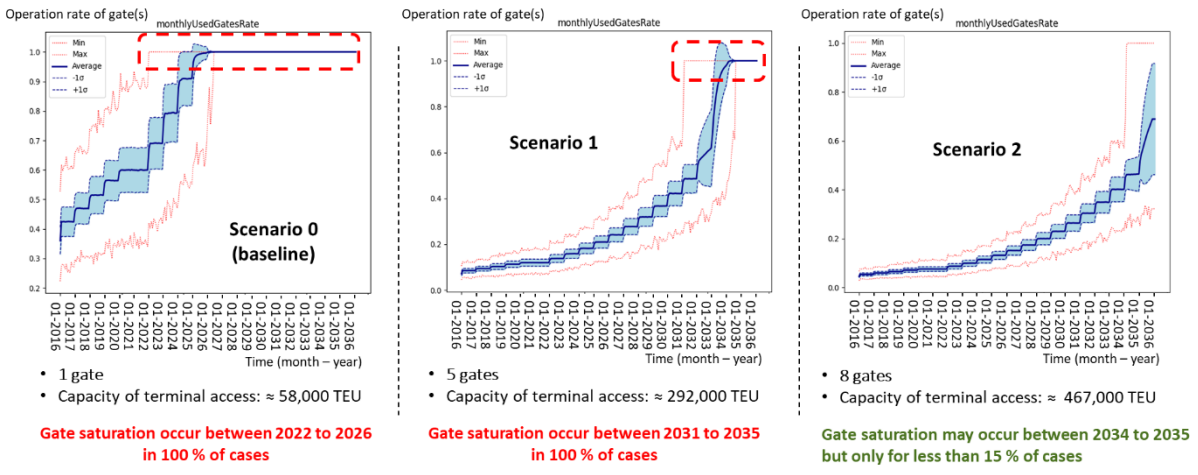


Figure 19 – Stochastic simulations of scenarios 0, 1 and 2

These final results shall of course be discussed with the port business users, who are in fact the actual final validators of the results obtained with the systemic digital twin.

4. Conclusion

In this short paper, we tried to present the motivations and the main features of our systemic digital twin approach, especially focusing on the design, construction and validation processes of such a tool, which are usually never highlighted in the literature. As one could see, our approach is based on a new formal system modeling language Σ which is highly innovative since it synthesizes systemic, stochastic, object-oriented, synchronous and parallel modeling paradigms which were never unified. We believe that this core choice brings fundamental benefits, both in terms of accuracy & correctness of the systemic digital twins developed with our approach, hence of business validity, and ease of evolution & maintenance, hence of associated operating recurring costs compared to traditional approaches (see for instance [43-44) since the Σ approach transforms the maintenance of a system digital twin into the maintenance of the underlying Σ model that can be managed using classical principles of software configuration management and maintenance (see [45-46] for more details).

One shall indeed not think that a systemic digital twin is a frozen tool. Any enterprise is a living body which permanently adapts, changes and transforms itself, in order to maintain its alignment with its business environment that evolves at its own independent pace. It is therefore key to integrate over time these evolutions within the systemic digital twin that was constructed in order to support continuous business improvements within the enterprise where it was deployed. It is only through such an intimate and long-term integration between the real business operations and the systemic digital twin that such a tool will provide the expected business value along time.

There are also of course many other specific aspects of the proposed approach that require further detail and elaboration. We would however finally like to stress the very crucial importance of systems architecture when constructing a systemic digital twin since this discipline enables the smooth integration of the various disciplines that are all providing a piece of the complex puzzle of a given industrial enterprise reality.

Acknowledgments

The two authors thank the anonymous referees for their valuable comments that allowed to improve the quality of this paper.

Role of authors

The two authors contributed equally in the conception, design, and drafting of the paper, in revising it critically for intellectual content, and in the final approval of the version to be published. All authors agree to be accountable for all aspects of the work

Disclosure of interest

There are no conflicts of interests to declare.

Fundings

This paper was financially supported by Systemic Intelligence Group.

References

1. Grieves M.W., *Product lifecycle management: the new paradigm for enterprises,* International Journal of Product Development, 2 (1-2), 71-84, 2005
2. Grieves M.W., *Digital twin: manufacturing excellence through virtual factory replication*, White paper, Digital Twin Institute, 7 pages, 2014
3. Grieves M.W., Vickers J., Dig*ital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems*, In "Transdisciplinary Perspectives on Complex Systems", 85-113, Springer
4. Schleich B., Anwer N., Mathieu L., Wartzack S., *Shaping the digital twin for design and production engineering*, CIRP Annals, 66 (1), 141-144, 2017
5. Jones D., Snider C., Nassehi A., Yon J., Hicks B., *Characterising the Digital Twin: A systematic literature review*, CIRP Journal of Manufacturing Science & Technology, 29 (A), 36-52,
6. Singh M., Fuenmayor E., Hinchy E.P., Qiao Y., Murray N., Devine D., *Digital Twin: Origin to Future*, Applied System Innovation, 4 (36), 2021
7. Stark R., Fresemann C., Lindow K., *Development and operation of Digital Twins for technical systems and services*, CIRP Annals, 68 (1), 129-132, 2019
8. Botín-Sanabria D.M., Mihaita A.-S., Peimbert-García R.E., Ramírez-Moreno M.A., Ramírez-Mendoza R.A., Lozoya-Santos J.d.J., *Digital Twin Technology Challenges and Applications: A Comprehensive Review*, Remote Sensing, 14, 1335, 2022
9. Matta A., Lugaresi G., *An Introduction to Digital Twins*, Proceedings of the Winter Simulation Conference (WSC'24), Orlando - Florida, 1281–1295, IEEE Press, 2025
10. Zhang M., Tao F., Huang B., Liu A., Wang L., Anwer N., Nee A.Y.C., *Digital twin data: methods and key technologies*, Digital Twin, 1 (2), 2022
11. Boschert S., Rosen R., *Digital Twin: The Simulation Aspect*, In "Mechatronic Futures", Hehenberger P., Bradley D. (eds), Springer, 2016
12. Qi Q., Tao F., Hu T., Anwer N., Liu A., Wei Y., Wang L., Nee A.Y.C., *Enabling technologies and tools for digital twin*, Journal of Manufacturing Systems, 58 (B), 3-21, 2021
13. Tao F., Anwer N., Liu A., Wang L., Nee A.Y.C., Li L., Zhang M., *Digital twin towards smart manufacturing and industry 4.0*, Journal of Manufacturing Systems, 58 (B), 1-2, 2021
14. Bernus P., Nemes L., Schmidt G., *Handbook on enterprise architecture*, Springer Science & Business Media, 2012
15. Winskel G., *The formal semantics of programming languages – An introduction*, The MIT Press, 1997
16. Batteux M., Prosvirnova T., Rauzy A., *From Models of Structures to Structures of Models,* IEEE International Symposium on Systems Engineering (ISSE 2018)*,* Roma, October 2018
17. Prosvirnova T., Batteux B., Brameret P.-A., Cherfi A., Friedlhuber T., Roussel J.-M., Rauzy A., *The AltaRica 3.0 project for Model-Based Safety Assessment*, In Proceedings of 4th IFAC Workshop on Dependable Control of Discrete Systems, DCDS'2013, York, Great Britain, 127–132, 2013
18. Hastings C., Rauzy A., *Foundations for Model-Based Systems Engineering and Model-Based Safety Assessment*, Systems Engineering, Vol. 22, Issue 2, 146–155, 2019
19. Sterman J.D., *Business Dynamics – Systems Thinking and Modeling for a Complex World*, Irwin McGraw-Hill, 2000
20. De Weck O., Krob D., Li L., Lui P.C., Rauzy A., Zhang X.G., *Handling the COVID-19 crisis: Toward an agile model-based systems approach,* Systems Engineering, Vol. **23**, Issue 5, 656-670, 2020

21. Blanchard B.S., Fabricky W.J., *Systems Engineering and Analysis*, Prentice Hall, 1998
22. Kossiakoff A., Sweet W.N., *Systems Engineering – Principles and Practice*, Wiley Series in Systems Engineering, 2003
23. Krob D., *Model-Based Systems Architecting – Using CESAM to architect complex systems*, ISTE and Wiley, 2022
24. Sage A.P., Armstrong J.E., *Introduction to systems engineering*, Wiley Series in Systems Engineering, 2000
25. Krob D., Rauzy A., *A Guided Tour of the Systemic Modeling Language* Σ, White Paper, Systemic Intelligence Group, 2022 (free access on www.systemic-intelligence.net/resources)
26. Marwedel P., *Embedded System Design*, Kluwer Academic Publishers, 2003
27. Batteux M. Prosvirnova T., Rauzy A., *System Structure Modeling Language (S2ML)*, HAL, 2015 – Access through https://hal.science/hal-01234903/
28. Krob D., *Model-Based Systems Architecting – Using CESAM to Architect Complex Systems*, 217 pages, ISTE & Wiley, 2022
29. Rauzy A., *Σ™ Reference Manuel,* 89 pages, Systemic Intelligence Group, 2024 (free access on www.systemic-intelligence.net/resources)
30. Rauzy A., *WIDL Reference Manuel,* 58 pages, Systemic Intelligence Group, 2024 (free access on www.systemic-intelligence.net/resources)
31. Solheim A.V., Rauzy A., Brett P.O., Ellefmo S., Hatling T., Helmons R., Asbjørnslett R.E., *Assessment of expected production of a deep-sea mining system: An integrated model-based systems engineering and discrete event simulation approach*, Systems Engineering, 26 (6), 847-858, 2023
32. Benkamoun N., ElMaraghy W., Huyet A.L, Kouiss K., *Architecture Framework for Manufacturing System Design*, Procedia CIRP, 17, 88-93, 2014
33. Saez M., Barton K., Maturana F., Tilbury D.M., *Modeling framework to support decision making and control of manufacturing systems considering the relationship between productivity, reliability, quality, and energy consumption*, Journal of Manufacturing Systems, 62, 925-938, 2022
34. Xu Q., Wang J., Gao W., et alii, *Digital Twin: State-of-the-Art and Future Perspectives,* In "ICCSMT '24: Proceeding of the 2024 5th International Conference on Computer Science and Management Technology", 731-840, 2024
35. Fuhrländer-Völker D., Lindner M., von Elling M. et allii, *Method for the development and application of digital twins in manufacturing,* Production Engineering Research and Development, 19, 959–973, 2025
36. ISO, ISO/IEC 30173:2023, *Digital Twin - Concepts and Terminology*, 2023
37. David I., Bork D., *Towards a Taxonomy of Digital Twin Evolution for Technical Sustainability*, in "ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion", 934-938, 2023
38. Dittler, D., Lierhammer, P., Braun, D., Müller, T., Jazdi, N., & Weyrich, M. , *A novel model adaption approach for intelligent digital twins of modular production systems*, In "2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA)", 1-8. IEEE, Sept. 2023
39. Fur S., Heithoff M., Michael J., Netz L., Pfeiffer J., Rumpe B., Wortmann A., *Sustainable Digital Twin Engineering for the Internet of Production*, In "Digital Twin Driven Intelligent Systems & Emerging Metaverse", 101-121, Springer, 2023

40. Lugaresi, G., Gangemi, S., Gazzoni, G., & Matta, A., *Online validation of simulation-based digital twins exploiting time series analysis*, In "2022 Winter Simulation Conference (WSC)", 2912-2923, IEEE, Dec. 2022
41. Dittler, D., Stegmaier, V., Jazdi, N., & Weyrich, M. , *Illustrating the benefits of efficient creation and adaption of behavior models in intelligent Digital Twins over the machine life cycle*, Journal of Manufacturing Systems, *76*, 520-539, 2024
42. Su C., Tang X., Jiang Q., et al**.,** *Digital twin system for manufacturing processes based on a multi-layer knowledge graph model,* Scientific Reports, 15, article 12835, 2025
43. Akcay A., Biller S., Gan B. P., Laroque C., Shao G., *Maintenance and Operations of Manufacturing Digital Twins*, In "*2023 Winter Simulation Conference" (WSC 2023),* 1888-1899, IEEE, 2023.
44. Muctadir H. M., Manrique Negrin D. A., Gunasekaran R., Cleophas L., Van den Brand M., Haverkort B. R. *Current trends in digital twin development, maintenance, and operation: an interview study*, *Software and Systems Modeling*, 1-31, 2024
45. Grubb P., Takang A.A., *Software maintenance: concepts and practices*, World Scientific, 2003
46. Quigley J.M., Robertson K.M., *Configuration Management: Theory, Practice, and Application*, CRC Press, 2015