Managing the Systemic Digital Twin of an Industrial Enterprise with WorldLab & Σ

Daniel Krob & Antoine Rauzy[1]

Systemic Intelligence Group ©

June 2022

Table of contents

[1] Emails of the authors: daniel.krob@systemic-intelligence.net and antoine.rauzy@systemic-intelligence.net

1. Introduction

A *digital twin* is a virtual representation of a system 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 and predict the temporal evolution of the considered system. Note that an enterprise shall be taken here in the meaning provided by enterprise architecture, 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. The concept of digital twin emerged during the last decade: to the best of our knowledge, it can be traced back to NASA which proposed in 2010 the first definition of a digital twin in an attempt to improve physical model simulation of spacecrafts [8].

Nowadays digital twins are integrated digital tools that allow the continuous improvement of a given product or process from a global or end-to-end perspective. They currently mainly correspond to either computer aided design tools (CAD), focusing on geometrical representations of a system, or to 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 CAD and MBSE capabilities are still very 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 in practice to silo-ed geometrical & functional digital twins, at the opposite of the idea underlying this concept.

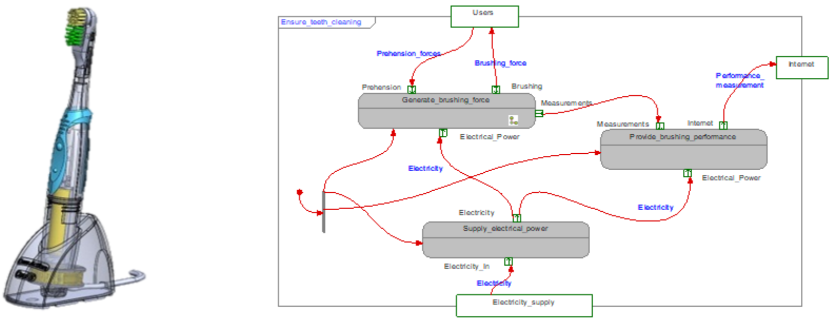

Figure 1 – Geometric and functional representations of a given system, here an electrical toothbrush

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 energy interactions with a given system. But geometry is quite badly adapted when one wants to deal with cost, quality, delay and/or business performances of a system, that can be much more easily addressed by a functional approach, which moreover could also be used as an entry point to physical simulations based on system geometrical representations. This explains why our approach to 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—in its mathematical meaning—semantics [12]: 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 as system. 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 formal approach to 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 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 WorldLab and 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 number of monthly users of a service or the number of annual buyers of a product, 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 [3].

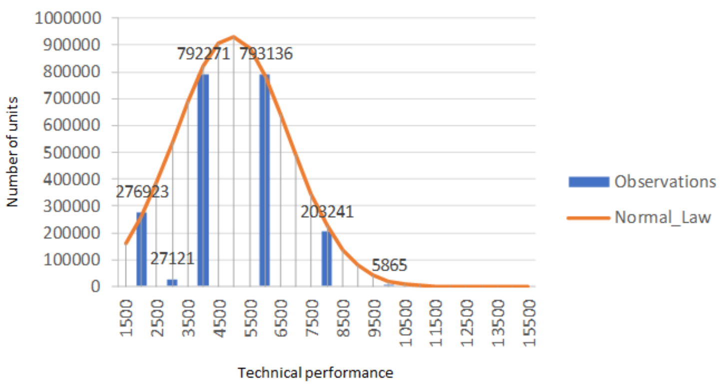

Figure 2 – Example of a Normal probabilistic modeling of the number of products sold per year by an industrial company (each blue bar corresponding to a different type of product of a same family)

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 will be 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. 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 short 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 two main sections, which are respectively dedicated to the specific principles of the so-called systemic digital twin approach that we are following and to the description of the systemic digital twin design, developing and use process that we propose, followed by a short conclusion highlighting the key benefits that are obtained in this way.

2. 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 in the context of enterprise modeling [11], that we shall now present.

2.1 Systemic digital twins

As already stated, we are indeed specifically proposing a *systemic digital twin* approach. 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 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 organization, maintenance strategy or service policy, 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 [1], [4], [5] or [10]), 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 [11] (see for instance Figure 3).

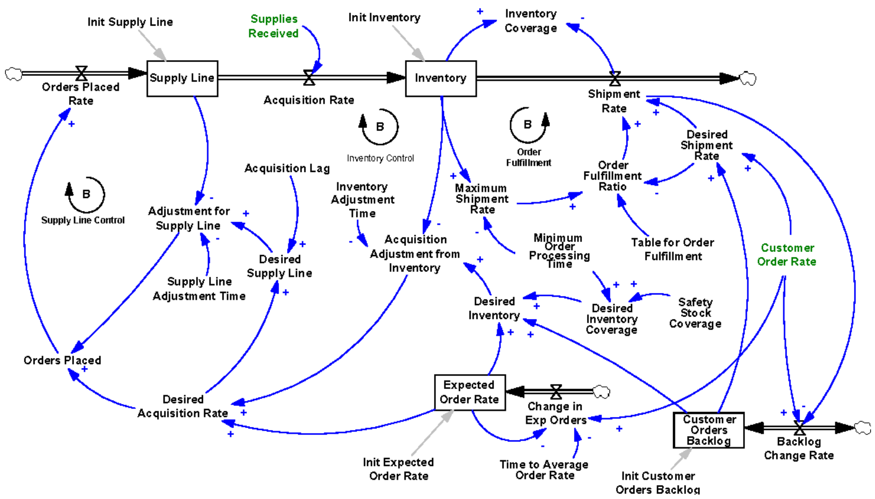

Figure 3 – Example of a systemic model, here for modeling a supply-chain strategy, proposed by Sterman

While the philosophy of our 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 not well 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 results [12] can then be more easily adapted in such a framework.

2.2 WorldLab and Σ

It is now time to introduce WorldLab and Σ which are the software tools on which our systemic digital twin approach is based. 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 the equivalent of a software engineering workshop for enterprise models instead of software programs. WorldLab indeed allows both to model an enterprise, based on the specific Σ modeling language that we will introduce soon, and to simulate immediately – based on a compiling approach—these enterprise models with Σ.

An overview of the WorldLab architecture is provided by Figure 4. This software platform has two main layers: the first is a data layer where natural environment, societal, governance, financial and economic data, continuously collected, transformed and organized from public and corporate data sources, are available for enterprise modeling & simulating purposes; the second layer specifically offers enterprise modeling & simulation capabilities, based on the Σ modeling language, where enterprise modeling specialists can, on one hand, develop enterprise models, reusing & adapting, when relevant, existing models, and, on the other hand, simulate these enterprise models in order to compute a number of predefined analytic syntheses that can be used to make business decisions. 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 kinds of principles than the 20-year-old AltaRica model-based safety technology (see [3] or [9]) from which it inherits both a number of key features and its industrial robustness.

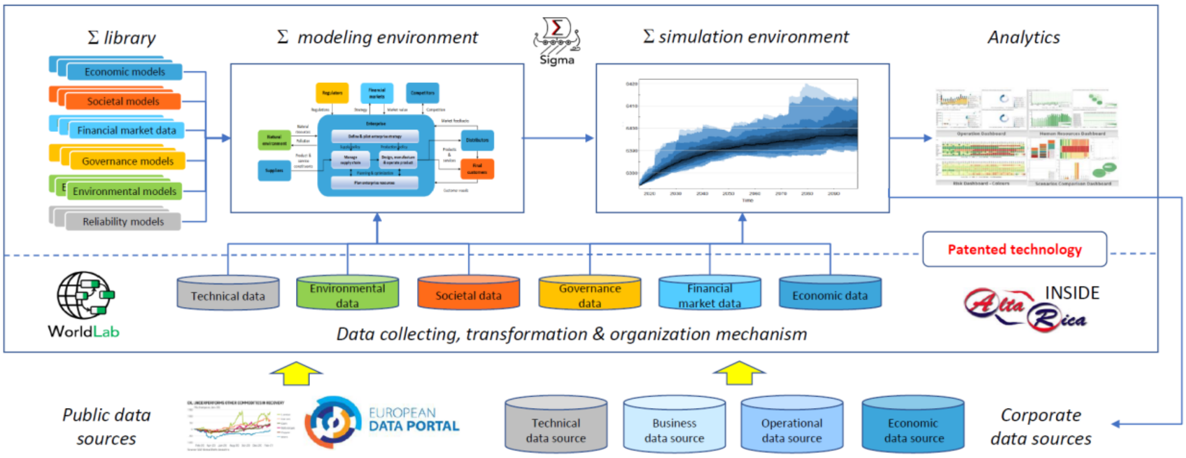

Figure 4 – WorldLab architecture overview

Note that the data and the models managed and proposed by the WorldLab platform are organized according to a standard world model that was initiated through the seminal paper [2]. In this specific world model, named according to the CESAM methodology [5] with which it was constructed, the world system is decomposed into five main sub-systems: 1) the natural environment formed of all natural resources that are involved in human activities, 2) the human population, who – by the way – is currently stressed by the covid-19 coronavirus, 3) the economical system formed of all economical entities in the world, 4) the governance system formed of all state political & regulatory entities on Earth, 5) the financial market that we will not consider as part of the economic system. Each of the main subsystems of the CESAM world model can be decomposed according to a standard breakdown structure as depicted in the below Figure 5, which provides a useful & relevant classification framework for enterprise modeling specialists. If we are limiting the analysis to the main actors which concentrate wealth and size, the human society does in particular not appear so complex since there are only around 100,000 key actors to take into account in order to capture and understand the world system dynamics! Note also that the main concrete actors (companies, countries, governing bodies, etc.) are appearing quite quickly at level 3 or 4 of our system breakdown.

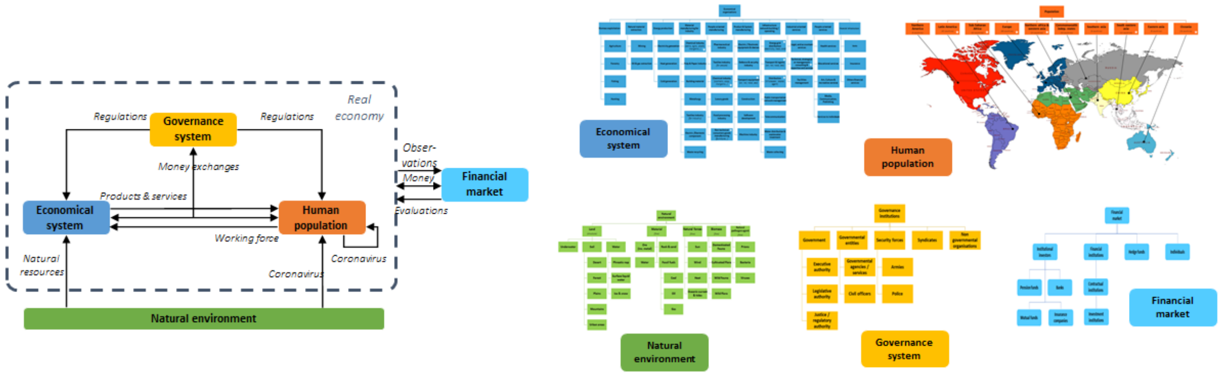

Figure 5 – Structure of the CESAM world model

The core part of WorldLab is of course the Σ language which is presented in details in [5]. 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 of course 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 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 [7]. 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.

To illustrate how Σ practically works, one can find in Figure 6, a very simple example of a model in Σ where we modelled the world gold ecosystem as the union of Earth, gold corporations and gold users. The Earth provides gold resources with an initial value of 50,000 tons of gold according to the current estimates. Gold corporations extract gold ore as long as there is gold on Earth and produce gold every year, if they have stocks of ore to do so, these two business processes – referred as "activities" in Σ – being operated in parallel within one year duration. Gold consumers buy gold at their own rhythm, which increases their stocks of gold, currently amounting to 190,000 tons, but of course they cannot buy more than available in producer stocks. Note that the Σ model provided in the previous figure models the annual production of gold and the new demands for gold through normal distributions, via the "normalDeviate" operator, having roughly similar distributions, as observed on the market, which allowed us to illustrate a typical use of the normal deviation operator. To be more realistic, we could of course introduce dedicated functions to estimate the evolution of gold supply and demand.

```
system GoldEcosystem
    system Earth … end
    system GoldCorps … end
    system GoldUsers … end
end

system GoldEcosystem.Earth
    float GoldResources (init = 5*e+4, unit = t);
end

system GoldEcosystem.GoldCorps
    float GoldCorpStocks (init = 0, unit = t);
    float GoldYearlyProduction (init = 0, unit = t);
    float GoldOreStocks (init = 0, unit = t);
    float GoldYearlyExtraction (init = 0, unit = t);
    bool ProducingGold(init = false);
    bool ExtractingGold(init = false);

   activity ExtractGold
     trigger:
       return main.Earth.GoldResources > 0 and ExtractingGold = false;
     start: {
       ExtractingGold = true;
       GoldYearlyExtraction = min(main.Earth.GoldResources, normalDeviate(2000,100));
       main.Earth.GoldResources = main.Earth.GoldResources –  GoldYearlyExtraction;  }
     completion:
        ExtractingGold = false;
        GoldOreStocks += GoldYearlyExtraction;
     duration:
       return 1; // year
   end

   activity ProduceGold
     trigger:
       return GoldOreStocks > 0 and ProducingGold = false;
     start: {
       ProducingGold = true;
       GoldYearlyProduction = min(GoldOreStocks, normalDeviate(1750, 150));
       GoldOreStocks = GoldOreStocks - GoldYearlyProduction; }
     completion: {
       ProducingGold = false;
       GoldCorpStocks += GoldYearlyProduction;
      }
     duration:
       return 1; // year
   end
end

system GoldEcosystem.GoldUsers
   float GoldUserStocks (init = 1.9*e+5, unit = t);
   float GoldYearlyNewDemand (unit = t);
   bool PurchasingGold (init = false);

   activity PurchaseNewGold
     trigger:
       return not PurchasingGold and GoldCorpStocks > 0;
     start: {
       PurchasingGold = true;
       GoldYearlyNewDemand = normalDeviate(1800,200);
       Main.GoldCorpStocks -= min(main.GoldCorpStocks, GoldYearlyNewDemand);   }
     completion: {
       PurchasingGold = false;
       GoldUserStocks += GoldYearlyNewDemand;
      }
     duration:
       return 1; // year
   end
end
```

Figure 6 – Example of a model in Σ

Note also that a choice had to be made whether the transaction was to be seen from a user point of view (purchasing) or a corporate point of view (sales), which both remain equivalent in terms of systemic modeling.

Finally, one can also see here the main difference between programming and modeling: 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 therefore still an art …

2.3 Systemic digital twin management process

Managing 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.

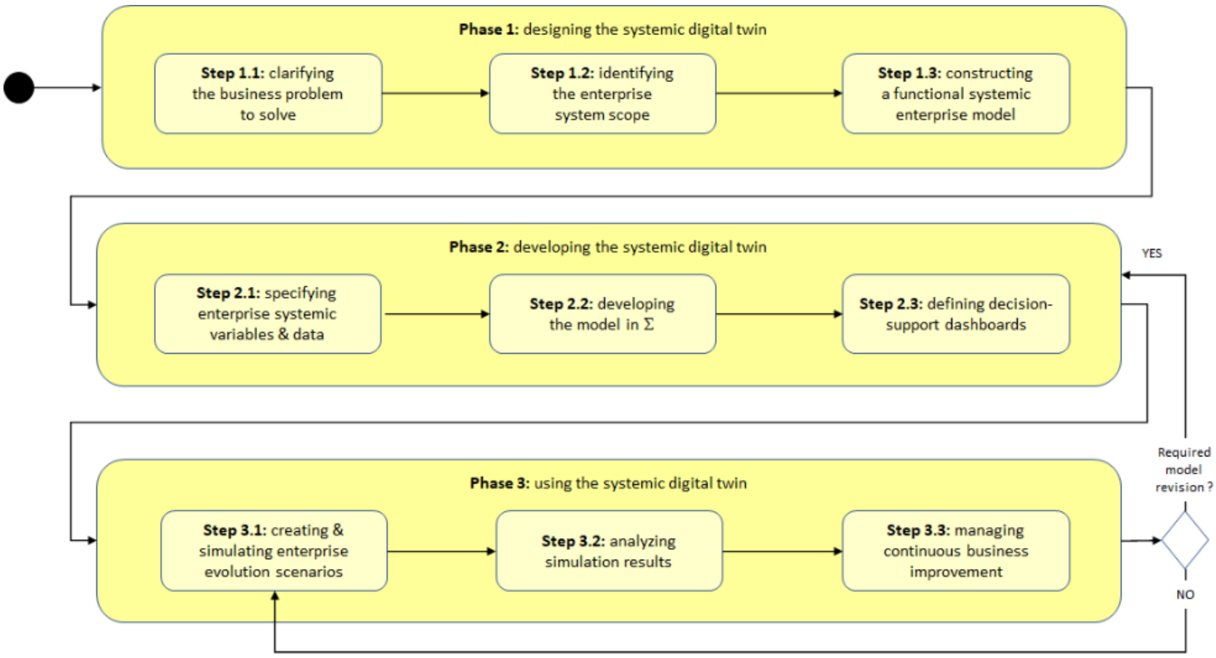

Figure 7 – Overview of the systemic digital twin management process

This digital twin management process has three main phases, respectively focused on designing, developing, and using 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 problem 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; the next steps consist then in 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 stabilized;
- *Phase 2 – development*: the second phase consists then in developing a model using the Σ language within the WorldLab platform: one shall here begin by using 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 digital twin, before creating the relevant enterprise decision-support dashboards & alerts that shall help to solve the initially identified business problem;

- *Phase 3 – use*: the last phase is finally a business phase where one shall first create & simulate the enterprise evolution scenarios to study, then analyze the simulation results & alerts and propose 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 creation and associated simulations, analyses & alerts, and possibly revisions of the model in Σ.

A synthetic presentation of this process can also be found in Figure 7. Note finally that the next section 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.

3. Designing, developing & using a systemic digital twin in practice

We shall now show how to design, develop and use in practice a systemic digital twin with WorldLab and Σ, through a case study that was constructed by anonymizing and simplifying a real industrial company 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 e-TB company which is a high-tech enterprise that produces, at demand for industrial customers within a B2B model, electronic toothbrushes which are formed of three logical components, that is to say a base, a head and a body, as described in Figure 8.

e-TB designs each toothbrush depending on the needs of its customers as captured by salespersons, sources then amongst pre-selected suppliers the base, head and body components that are answering to these needs, and assembles afterwards all these components altogether on its assembly line, before delivering the final products to its customers. Since e-TB is highly committed on having a time-to-delivery delay as short as possible, marketing tries to anticipate the volumes of components that will be required per quarter, based on an analysis of market data and real sales & manufacturing data. Based on this marketing analysis, anticipated base, head and body components are then bought in advance and stored within e-TB warehouses. This policy makes components available, with only a 1-day delay, when a customer request has to be fulfilled.

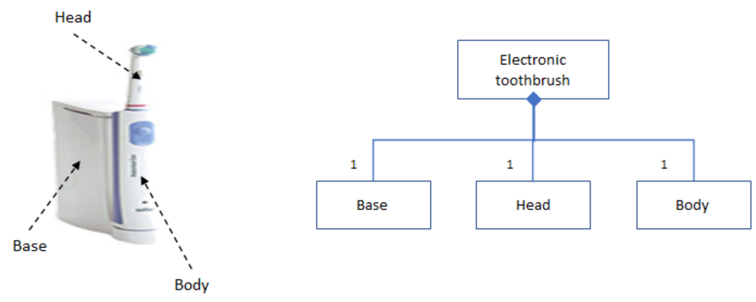

Figure 8 – Logical structure of the electronic toothbrushes produced by e-TB

However, this strategy may fail when marketing anticipations are not accurate. In this situation, there are then uncompressible delays for producing from scratch the missing specific components that are answering to the actual customer needs, since they were not stored. In such context, the associated sourcing delays – including storing – per component are provided in the table below.

Component	Sourcing & storing delay
Base	2 months
Head	1 month
Body	1 month

Figure 9 - Sourcing & storing delays per component when one has to produce them from scratch

These elements of context being recalled, let us now see how an associated systemic digital twin can be designed, developed, and used to support e-TB time-to-delivery optimization strategy.

3.1 Phase 1: designing the systemic digital twin

The very first phase towards a systemic digital twin for e-TB is the design phase. Its purpose is to scope the future e-TB systemic digital twin by constructing a first informal model of e-TB, which shall be challenged & validated before going to the next Σ modeling phase. The key point to have in mind is that it is not possible to go to the formal modeling phase before having stabilized our understanding of e-TB, which may take some time due to the many interactions and feedbacks that one needs to do with and to capture from e-TB actors. Since the e-TB model will fundamentally be unstable during this phase, we are thus using here informal modeling tools which are well adapted to this situation.

Step 1.1: clarifying the business problem to solve

The starting point of our process consists in clarifying the business problem that e-TB wants to solve. 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.

In our case, it happens that e-TB is a time-to-delivery oriented company as already stated. However, we could see in the introduction of this section that the way e-TB is organized in order to achieve the best possible time-to-delivery of its products is not optimal since delivery time can just be very bad, up to 2 months of delay, when marketing anticipations are not accurate, resulting in the lack of some key components in e-TB warehouses that may however be mandatory in order to answer to some specific customer needs. Moreover, the e-TB time-to-delivery policy has also a bad side effect: e-TB is indeed obliged to overstore the components in its warehouses in order to increase the probability of finding there the right components when a given customer requires them, knowing experimentally that component figures anticipated by the marketing are wrong. In other words, e-TB buys delivery time with storing space. But this policy unfortunately results in lots of overstored electronic toothbrush components at the end of each year. e-TB must then regularly destroy them since they are quickly not anymore adapted to the market which evolves at a very rapid pace in the e-TB high tech business. The cost of such over storage, mainly due to the difficulty of having accurate marketing predictions, is quite high and reaches more than 10 million euros per year, which are just lost each year by e-TB. It is thus quite key to find the best possible balance between time-to-delivery and levels of warehouse stocks.

Consequently, we can see that the business problem that e-TB has to solve is to find the best way of working that will allow the company to *minimize both the customer delivery time and the component storage volume of its electronic toothbrushes*. We were therefore able to express the business problem of e-TB in terms of an optimization problem, which clarifies it quite well.

Step 1.2: identifying the enterprise system scope

The second step of the design phase consists in creating a system vision of the e-TB company. The associated deliverable shall thus be a system logical interaction diagram for the considered enterprise, i.e., a diagram describing both the system / logical breakdown of the perimeter of e-TB which matters for the problem that we want to solve as identified in step 1.1, the associated relevant stakeholders, and the internal & external exchanges that they are managing. The resulting typical deliverable is presented in Figure 10 for the e-TB case study.

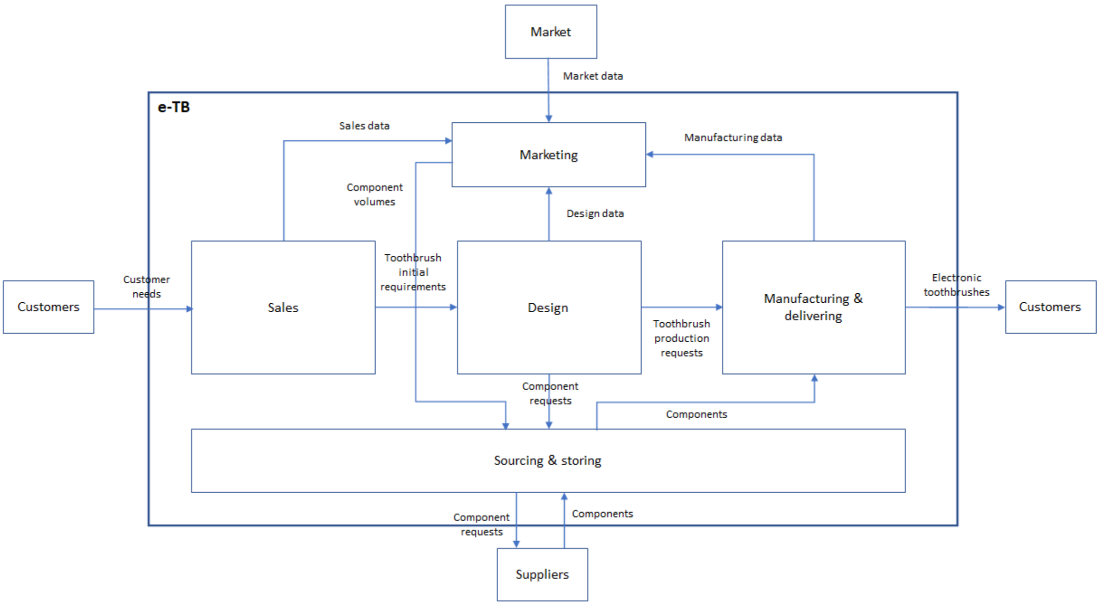

Figure 10 – System logical interaction diagram for e-TB company

This deliverable is obtained by identifying first the main organizational parts of the e-TB company that have to deal with the time-to-delivery and storage optimization problem that we have to address. The first obvious such parts can be easily obtained by analyzing the e-TB customer to customer end-to-end delivery process: this process starts with a sales part where the sales division of e-TB captures the customer needs, as stated by the customer. Then, it transforms these needs into toothbrush initial requirements. The latter may however not be totally complete/satisfactory from a technical point of view. The process then goes on by a design part where a product designer from the e-TB design division analyses the toothbrush initial requirements and defines the toothbrush final technical requirements, taking especially into account the current availability of components. This may lead to replacing ordered components by similar components, leading then both to component requests to the sourcing & storing division of e-TB and to toothbrush production requests to the e-TB manufacturing & delivery division. The latter will eventually deliver the expected electronic toothbrushes to the customer.

To complete the analysis, one has of course to integrate in the picture the marketing division of e-TB which plays a key role since it shall send each quarter anticipated component volumes to the sourcing & storing division of e-TB, based initially on market data at the beginning of each year and updated each quarter based on the actual sales, design & manufacturing data. Note also that one shall not forget the external stakeholders, here customers, market and suppliers, with which e-TB has external exchanges. In this matter, the new exchanges that one must add to complete the picture are just the exchanges – of component requests & components – that the sourcing & storing division of e-TB has with its suppliers and that we did not trace up to now (see Figure 10 for the final deliverable).

Step 1.3: constructing a functional systemic enterprise model

The next and last step of the design phase consists in enriching the enterprise vision that was captured in the previous step with a functional point of view. In other words, it consists in refining the previous analysis by identifying all the business processes associated with the various e-TB divisions, that are relevant with respect to the business problem that we want to address here. The associated typical deliverable is illustrated in Figure 11 for the e-TB case.

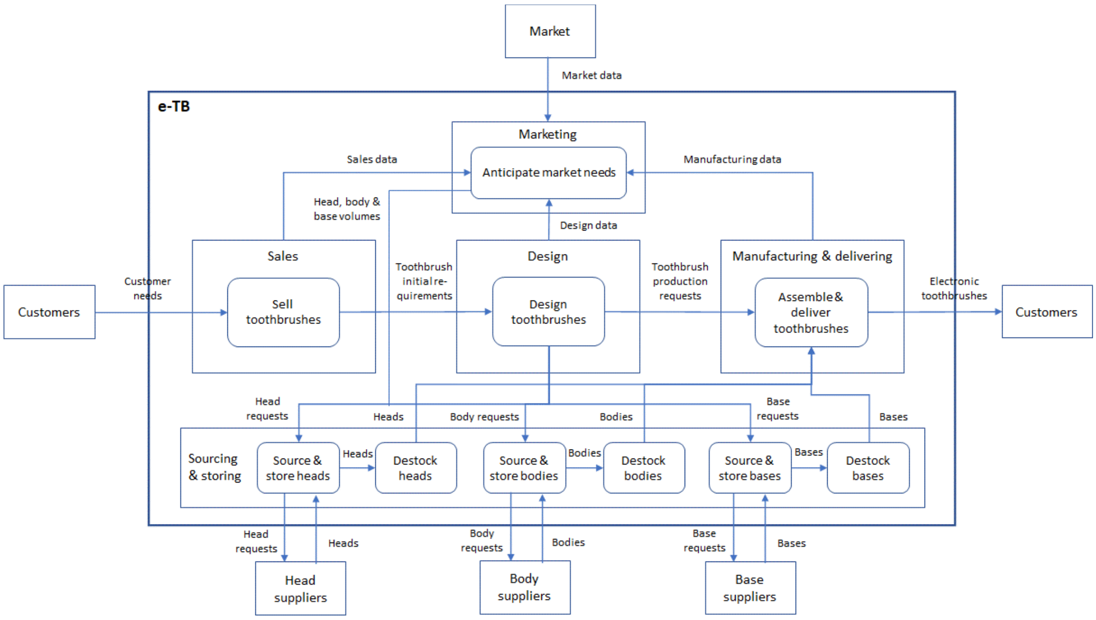

Figure 11 – Functional systemic model of e-TB aligned with its system breakdown

As one sees, we just completed the system logical interaction diagram obtained in step 1.2 by eliciting the underlying business processes. In our case, most of the e-TB divisions that matter are only involved through one single business process: "anticipate market needs" for the marketing division, "sell toothbrushes" for the sales division, "design toothbrushes" for the design division and "assemble & deliver toothbrushes" for the manufacturing & delivering division. In this last matter, note that we grouped together the "assemble toothbrushes" and "deliver toothbrushes" business processes since the deliver process – which is quite efficient and stable in the e-TB case—has practically no impact at all, both on the time-to-delivery durations and on the storage volumes. Finally, we were naturally led to introduce six business processes within the sourcing & storing division of e-TB: on a first hand, "source & store heads", "source & store bodies" and "source & store bases" since these processes are of course independent and, as so, each of them can have a different & potentially huge impact on the two key performance indicators – time-to-delivery and storage volume—of interest and, on a second hand, the associated destocking processes, i.e. "destock heads" "destock bodies" and "destock bases".

At this point, we now have a complete vision from an overall system perspective – even if still quite informal – of the key e-TB architectural static elements, both in terms of stakeholders, main internal enterprise divisions, internal & external exchange flows and business processes, with respect to the time-to-delivery and component storage volume optimization problem that we shall address.

3.2 Phase 2: developing the systemic digital twin

The second phase consists in developing the systemic digital twin of e-TB using WorldLab and Σ. This core phase is the most technical since one has here to manage both the variables & data, the formal model and the decision-support dashboards & alerts that are the key constituents of the e-TB systemic digital twin. Note that one may need to come regularly back to this phase for managing evolutions or adaptations of the e-TB systemic digital twin in a continuous improvement business perspective.

Step 2.1: specifying enterprise systemic variables, data and activities

In this second development phase, the very first step is dedicated to the identification & calibration of all variables & data that will be managed by the e-TB systemic digital twin. We need in particular first to understand more in details the exchanged variables that appeared—and were abstracted—in the functional systemic enterprise model defined in the last previous step.

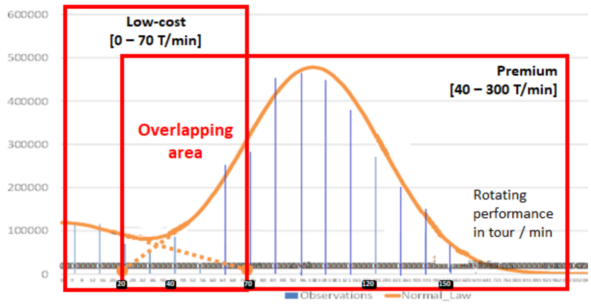

Figure 12 – Annual e-TB electronic toothbrush sales per rotating performance

Let us start therefore by looking more precisely on the electronic toothbrushes that are manufactured, as a main mission, by e-TB, since all the flows exchanged between the various e-TB business processes identified in step 1.3 are dependent of the nature of the produced electronic toothbrushes. In this last matter, it happens that e-TB is in fact producing two different types of electronic toothbrushes, the low-cost ones and the premium ones, that one can in particular technically differentiate through the range of their rotating speeds, which have their own independent business dynamics. One can indeed see from the annual sales statistics of e-TB (see Figure 12) that the sales distributions of low-cost and premium products per rotating speed are quite different, the first following a normal truncated law when the second has a usual normal distribution. Consequently, we must introduce specific variables that are taking into account this situation. As an example, one can typically see that the total amount of electronic toothbrushes produced by e-TB per unit of time is not a good systemic variable, since it abstracts too much the reality of the e-TB business, even if it is a good performance indicator for e-TB: this variable is indeed the sum of the two similar variables, respectively associated with low-cost and premium products, which are therefore the relevant systemic variables to consider. Note also that one will have to introduce in the same way specific variables for describing the components that are specifically used, either for low-cost, or premium electronic toothbrush construction.

One can now pass to the identification of the systemic variables associated with e-TB. In this matter, one has to review the main systems involved in the e-TB ecosystem – which groups both the market, the customers and all types of suppliers – and the marketing, sales, design, sourcing & storing and manufacturing & delivering divisions of e-TB, as synthetized in phase 1, in order to define the relevant systemic variables associated to these various ecosystem & enterprise systems. The result of this systemic variable review for e-TB is provided in the below table (see Figure 13).

The next step will then be to identify, when relevant, the initial values of these systemic variables and the data sources from which they can be taken, again when relevant. Here most of the e-TB systemic variables are triggered by customer demands that are not fixed: they shall then have a zero initial value which means that we shall only consider and measure them from the start of the systemic digital twin simulation process. The other variables that may be initialized in a different way are the quarterly anticipated electronic toothbrushes and the number of stored components, that we shall put to the initial values of these variables at the date chosen as the reference starting date of the systemic digital twin life. In terms of data sources, one has of course to know where one can find these data in order to be able to compare and to regularly reconciliate their estimated and actual values.

System	Systemic variable
Customers	Low-cost ordered toothbrushes Premium ordered toothbrushes
Marketing	Low-cost quarterly anticipated toothbrushes Premium quarterly anticipated toothbrushes
Sales	Low-cost sold toothbrushes Premium sold toothbrushes
Design	Low-cost designed toothbrushes Premium designed toothbrushes
Sourcing & storing	Low-cost stored body components Premium stored body components Low-cost stored head components Premium stored head components Low-cost stored base components Premium stored base components
Manufacturing & delivering	Low-cost manufactured toothbrushes Premium manufactured toothbrushes

Figure 13 – Systemic variables associated with all systems involved in e-TB company ecosystem

Systemic variables can be seen as stocks. Each actor owns a number of stocks. For instance, the sub-system Customers owns two stocks of ordered toothbrushes (one for low-cost, the other one for premium toothbrushes), the sub-system Marketing owns a stock of low-cost quarterly anticipated toothbrushes and so on. The notion of stocks should thus be taken here in a broad sense, i.e. quantities of interest for the business processes under study.

Activities of the various actors modify these stocks. In order words, they describe flows between stocks. For instance, the activity "sell toothbrushes" of the sub-system Sales increases its stocks of sold low-cost and premium toothbrushes, accordingly to the increasing of the stocks of ordered low-cost and premium toothbrushes of the sub-system Customers: in other words, it is synchronized with the activity "buy toothbrushes" of the sub-system Customers that we may also introduce.

At this point, several remarks should be made.

First, some activities are constrained by the availability of material or organizational resources. For instance, the production capacity of the provider(s) of toothbrush components may be limited. Such quantities are not variables *stricto sensu* as their values do not change during a given simulation (but may change from one simulation to the other). Rather, they are parameters. These parameters are specifically monitored as so, both in Σ and at WorldLab platform level.

Second, the result of activities may not be fixed, but rather vary according to some statistical distribution. For instance, the quantity of low-cost and premium toothbrushes sold every day by the Sales department may vary depending on the season or within the month. In Σ, there are two ways to describe such distributions: either by means of built-ins encoding of some widely used distributions such as – for instance – the normal, exponential, or Weibull distributions, or by means of externally defined distributions, so-called empirical distributions, concretely given in external files as set of points. Empirical distributions provide a direct link with field data.

Third, as already pointed out, activities of the various actors are performed in parallel, possibly at different paces. Each activity has thus its own duration. Consequently, one shall not forget to elicit the time features of each e-TB ecosystem business process (see the table of Figure 14 for the results in the e-TB case of this last activity). Note that no duration values are provided for the "buy toothbrushes" process since we do not have insight on it. However, it is not so important since each "buy" process of the customers will be synchronized with the associated "sell" process at e-TB level, which of course has also to be captured in Σ. As for results of activities, durations can be either deterministic or obey some statistical distribution.

Business process	**Duration delay**	**Duration unit**
Buy toothbrushes	Non available	Non available
Anticipate market needs	1	quarter
Sell toothbrushes	normal-deviate(10,5)	day
Design toothbrushes	normal-deviate(5,2)	day
Assemble & deliver toothbrushes	5	day
Source & store heads	1	month
Destore heads	1	day
Source & store bodies	1	month
Destore bodies	1	day
Source & store bases	2	month
Destore bases	1	day

Figure 14 – Durations of e-TB ecosystem business processes

Fourth, each activity is started when, or more exactly as soon as, a certain condition on stocks is verified. This for two reasons: first, there may be not enough available resources, or on the contrary too many products, to start an activity. For instance, there is no need for the Marketing to buy new toothbrushes components if the stocks are filled enough. Second, once an activity started, it is in general not possible to start a new instance of the same activity until the activity is not completed. The actor can be seen here as the resource that is required to perform the activity. Describing the triggering condition of each activity is thus an important part of the modeling process.

Fifth, given what we just said, activities modify the state of the system – i.e. in a first approximation the values of stocks – twice: when they are started, to mobilize the resources they need to be performed and at their completion, as the result of their execution.

The description of stocks and activities, and for activities their triggering conditions, actions at start and at completion, and durations, is the process by which the informal model is transformed into a formal one. Once this description is achieved, writing down the Σ model becomes, to some extent, "just a matter of technique". This is not completely true however as the process by which both the informal and the formal models are designed are iterative and may depend on performance issues of assessment tools. For instance, it may be tempting to iterate an activity 24 times a day for the sake of accuracy, but in practice it may be better to consider it as a single daily activity. Moreover, Σ is a full-fledged prototype- and object-oriented language. Consequently, it provides elegant and efficient ways of representing concepts, which ease the authoring and the maintenance of systemic models. It is thus often the case that the Σ model is not a bare encoding of the informal model. Rather, it helps to structure further the analysis and to capture commonalities within the informal model and with previously developed models. In a word, the translation of the informal model into a formal one leads often to refining the informal model.

Step 2.2: developing the model in Σ

We now have everything in hands for constructing the model for e-TB in Σ. The skeleton of such a formal model can be first quite obviously derived from the informal model elaborated during the first phase and from the various elements defined in the previous step 2.1. The beginning of the skeleton of the e-TB formal model in Σ is provided in Figure 15. At this stage, it is just – as one can see here – a syntactic translation in the Σ language of the modeling material that we obtained so far, activities excluded. The objective of this first phase is to allocate systemic variables (stocks) to sub-systems and to check that the overall architecture of the system has been correctly captured.

```
system eTBEcosystem
   system Customers ... end
   system Sales ... end
   system Design ... end
   system SourcingStoring ... end
   system ManufacturingDelivering ... end
   system Marketing ... end
end

system eTBEcosystem.Customers
    int lowCostOrderedToothbrushes(init = 0);
    int premiumOrderedoothbrushes(init = 0);
end

system eTBEcosystem.Sales
    int lowCostSoldToothbrushes(init = 0);
    int premiumSoldToothbrushes(init = 0);
end
 ...
```

Figure 15 – Skeleton of the e-TB model in Σ

We are now ready to develop the full model in Σ of our enterprise. In the e-TB case, this shall lead us to develop formal descriptions of the behaviors of each part of the e-TB ecosystem, as defined in the very first breakdown assertion of the model in Σ (see Figure 15). As an illustration, Figure 16 shows the full description of the sub-system Sales and its activity “SellToothbrushes”.

```
system eTBEcosystem.Sales
    int lowCostSoldToothbrushes(init = 0);
    int premiumSoldToothbrushes(init = 0);
    bool selling(init = false);
    parameter float meanSaleDuration = 10;
    parameter float stddevSaleDuration = 5;

    activity SellToothBrushes
      trigger: return not selling;
      start: selling = true;
      completion: {
        int lowCostVolume, premiumVolume;
        lowCostVolume = empiricalDistribution(“Data/LowCostToothBrushesSalesData.csv”, linear);
        premiumVolume = empiricalDistribution(“Data/PremiumToothBrushesSalesData.csv”, linear);
        main.Customers.lowCostOrderedToothbrushes += lowCostVolume;
        main.Customers.premiumOrderedoothbrushes += premiumVolume;
        lowCostSoldToothbrushes += lowCostVolume;
        premiumSoldToothbrushes += premiumVolume;
        selling = false;
        }
      duration: min(1, normalDeviate(meanSaleDuration, stddevSaleDuration)); // days
    end
end
```

Figure 16 – Full description of the sub-system Sales in Σ

This model fragment intends here to capture the buy-sales relationships within e-TB ecosystem. One can in particular observe that it refers to two empirical distributions and one pair (meanSaleDuration, stddevSaleDuration) of parameters. The empirical distributions, which are loaded from .csv files, describe the volumes of low-cost and premium toothbrush sold over the time. These two datasets are typically obtained by experience feedback, possibly corrected by business expert judgement. The two parameters (meanSaleDuration, stddevSaleDuration) are used as inputs for a given normal distribution modeling the duration of the sale process. They can be modified prior to each simulation, so to study the impact of variations in the duration of the sale process. One can finally note that Boolean variables are introduced as triggering conditions of every activity in order to ensure that there cannot be two of the same activity occurring at the same time.

Note that developing the models in Σ is not an easy task. The Σ language is indeed highly constrained in order to oblige the modeling expert to think in the right systemic way. Moreover, it is usually when one arrives at this step that one will discover lots of gaps and missing items in the previous analyses, which shall be seen as a key benefit of our formal approach. Σ does indeed not allow neither ambiguity, nor approximations. Consequently, feedbacks of this new step on all the previous steps of the systemic digital twin construction process are just normal. They shall be always welcomed by all the modeling actors since they are key for constructing a robust systemic model.

Step 2.3: defining decision-support dashboards

Finally, the third and last step of the development of the systemic digital twin of e-TB using WorldLab and Σ consists in defining – through a dedicated WorldLab interface – the decision-support dashboards and alerts that will synthesize / manage the results of the simulations of our systemic digital twin.

In the e-TB case, the key performance indicators that we are interested in are the monthly distributions of stored component volumes per type of components and time-to-delivery durations per type of toothbrushes. Examples of possible associated decision-support dashboards are provided in the below Figure 17. Note that the time-to-delivery is measured on the right-hand side diagram in a relative way: e-TB indeed commits always on a given time-to-delivery with its customers during the sales phase, which leads the company to measure the difference between the actual and the committed time-to-delivery, which can be negative when e-TB is in advance or positive when e-TB is late, corresponding to two dynamics with respect to time-to-delivery expressed in the two curves of our diagram.

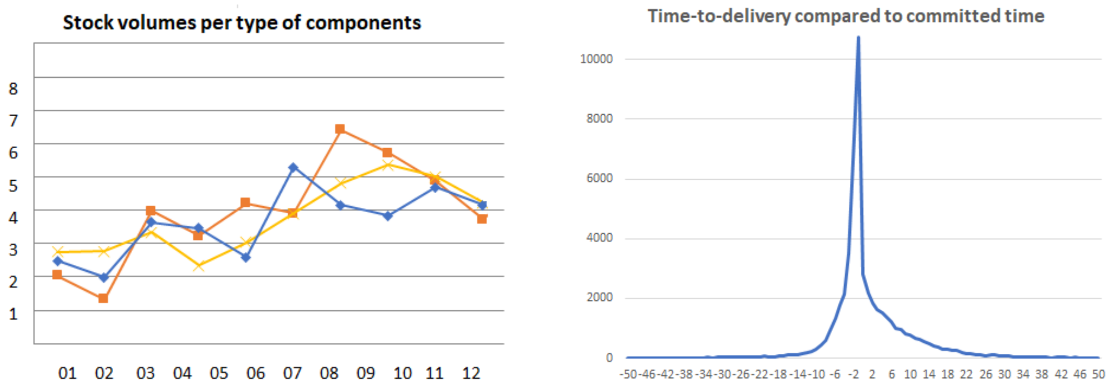

Figure 17 – Examples of decision-support dashboard for e-TB

Note finally that such decision-support dashboards are of course computed through simulations of the previous model developed in Σ, where one needs therefore to put specific measurement instructions for describing the measures to do during a simulation in order to feed these dashboards. Similar way of working may also be done with respect to alerts that the systemic digital twin may manage.

3.3 Phase 3: using the systemic digital twin

Finally, the third and last phase towards a systemic digital twin for e-TB is the use phase. We are here coming back to the business in order to help e-TB managers to take the right decisions, based on the analyses of the e-TB evolution scenarios provided by the systemic digital twin simulations, in order to solve the business problem identified at the very beginning of the process. This phase ends by installing the e-TB systemic digital twin as a standard operational tool within the organization that shall support continuous improvement of the enterprise on the initial business problem scope.

Step 3.1: creating & simulating enterprise evolution scenarios

The point is now to begin to use the systemic digital twin that we just constructed in order to solve the initial business problem and to improve the business operations. In the e-TB case, we know that the core problem to solve is to anticipate better the volumes of each body, head and base component involved in the manufacturing of an electronic toothbrush. This is why e-TB has to start by simulating the quarterly volumes of each component of its electronic toothbrushes. Such simulations are then managed by instantiating the free parameters of the e-TB Σ model to different values, leading to different simulation scenarios. One can typically simulate in this way the consequences of various potential behaviours of the electronic toothbrush market for the e-TB company.

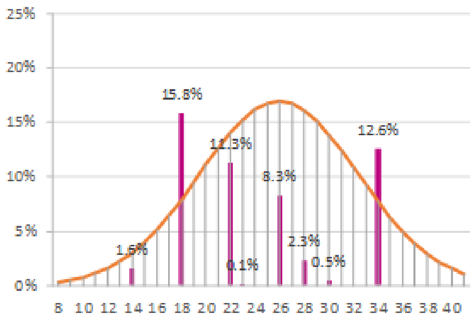

Figure 18 – Simulated quarterly sold volumes of an e-TB electronic toothbrush component

Figure 18 shows the result of such a simulation for the sold volumes of an e-TB electronic toothbrush component during the first quarter of a year, here expressed in percent of the total yearly amount of sold components, classified per increasing technical performance of the considered component, under the assumption of a normal market behavior. We also indicated on that figure the normal distribution that approximates the obtained simulation results.

Step 3.2: analyzing simulation results

The second step of the use phase consists in business analyses of the simulation results obtained in the previous step. The systemic digital twin only plays here a supporting role since this new step is fundamentally a business step which cannot be magically managed by a pure tooling approach. To understand better this point, one may have a look on the below Figure 19 which illustrates a typical business analysis that can be done on the basis of the simulation results provided by the last step, as synthesized by Figure 18 in the e-TB case.

One can indeed see that when one clusters the considered components according to the clusters that are highlighted in red in the left-hand side figure of Figure 18, the distribution of the clusters that one obtains in this way are perfectly captured by a normal distribution. This observation may for instance suggest rationalizing the e-TB component portfolio by choosing a reference component in each identified cluster that will replace all existing components of a given cluster, leading for instance here to 3 reference components in the example of Figure 19. Such a choice would indeed allow to cover the market – which is driven by the component performance, used here as the core analysis axis – with only 3 components instead of 6, which is a lever for both reducing the cost of these components since having less components allows to get bigger volumes and lower prices from the component suppliers and complexity of the corresponding component portfolio which would here be divided by 2.

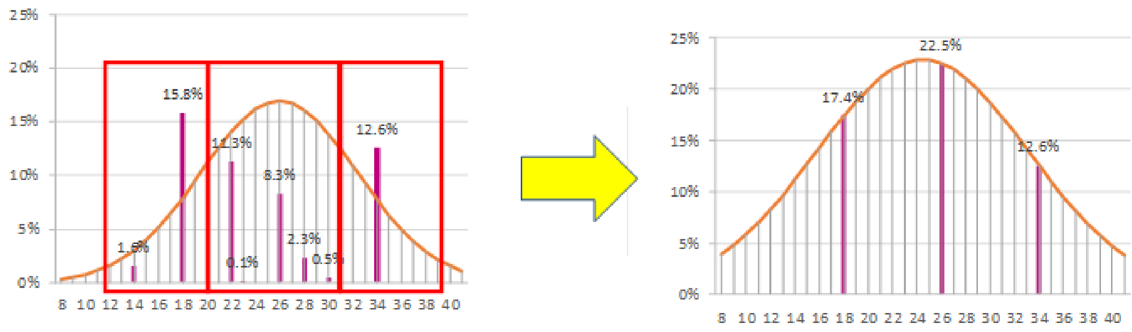

Figure 19 – Possible rationalization of the analyzed electronic toothbrush component

As one can see, such an analysis is clearly a business analysis where the systemic digital twin "just" provides the material to support such an analysis.

Step 3.3: managing continuous business improvement

Last, but not least, one may now integrate the systemic digital twin in the usual operational activities of the e-TB company. In such a case, it means that one shall use the systemic digital twin to define the best storage and time-to-delivery strategies and to monitor such strategies among time, in order to achieve optimal stored component volumes and electronic toothbrush time-to-delivery. The below Figure 20 shows for instance the comparison between the simulated (in red) and the actual (in green) monthly sold volumes of an e-TB electronic toothbrush component among time. The relatively good accuracy between estimation & reality – which is key to optimally manage the stocks of this component – is of course obtained through a regular re-calibration of the systemic digital twin among time.

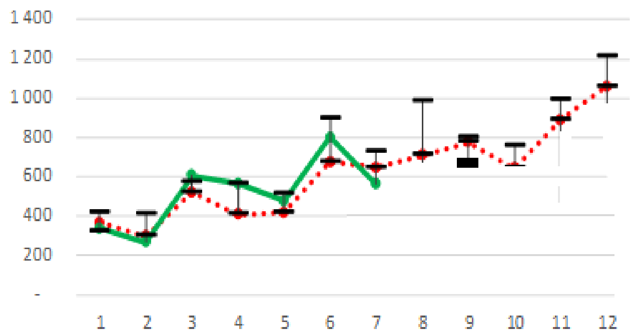

Figure 20 – Comparison of simulated & actual monthly sold volumes of an e-TB electronic toothbrush component

One shall thus not think that a systemic digital twin is a frozen tool. Any company is indeed a living body which permanently adapts, changes and transforms itself, in order to maintain its alignment with its business environment which 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 company 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.

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, development and use processes of such a tool, which are usually never highlighted in the literature. As one could see, our approach is based on a formal modeling language – Σ – in the line of the classical AltaRica language for safety (see [3] and [9]). We believe that this core choice brings fundamental benefits to the companies that use our approach, both in terms of accuracy & correctness of their systemic digital twins, hence of business validity, and ease of evolution & maintenance, hence of associated operating recurring costs.

There are of course many specific aspects of the proposed approach that require further detail and elaboration. We would 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 enterprise reality.

References

1. Blanchard B.S., Fabricky W.J., *Systems Engineering and Analysis*, Prentice Hall, 1998
2. 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
3. Hastings C., Rauzy A., *Foundations for Model-Based Systems Engineering and Model-Based Safety Assessment*, Systems Engineering, Vol. **22**, Issue 2, 146–155, 2019
4. Kossiakoff A., Sweet W.N., *Systems Engineering – Principles and Practice*, Wiley Series in Systems Engineering, 2003
5. Krob D., *Model-Based Systems Architecting – Using CESAM to architect complex systems*, ISTE and Wiley, 2022
6. Krob D., Rauzy A., *A Guided Tour of the Systemic Modeling Language* Σ, White Paper, Systemic Intelligence Group, 2022
7. Marwedel P., *Embedded System Design*, Kluwer Academic Publishers, 2003
8. Negri E., *A review of the roles of Digital Twin in CPS-based production systems,* Procedia Manufacturing, **11**, 939–948, 2017
9. 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
10. Sage A.P., Armstrong J.E., *Introduction to systems engineering*, Wiley Series in Systems Engineering, 2000
11. Sterman J.D., *Business Dynamics – Systems Thinking and Modeling for a Complex World*, Irwin McGraw-Hill, 2000
12. Winskel G., *The formal semantics of programming languages – An introduction*, The MIT Press, 1997