Modeling of Complex Systems: A minimalist and unified semantics for heterogeneous integrated systems

Boris Golden

PhD Candidate @ LIX (Ecole Polytechnique, France)

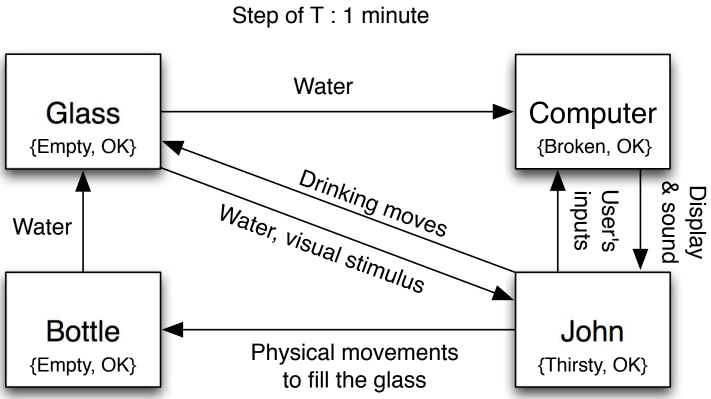
jointly with: Marc Aiguier (ECP), Daniel Krob (X)

December the 13^{th}, 2010

Introduction: what are we talking about ?

- Aim: modeling the functional behavior of real systems + their integration
- *real system*: set of interconnected elements with an overal functional behavior
- *system*: a functional mathematical object, that can be constructed by recursive integration of smaller systems
- in Systems Engineering, *complexity* comes from:
  - the *heterogeneity* of systems (physical, hardware, human)
  - the *integration* of systems (i.e. recursive applications of: composition + abstraction)
- Systems Engineering: conceptual and technical difficulties when building huge modern industrial systems, → industries needed to create complex engineering processes being able to involve a huge number of engineers coming from many different domains.

Introduction: systems approach



→ all the concepts introduced here are informal. We give a semantics to all the objects of this intuitive "graphical" language.

Time references (to define time)

Definition (Time reference)

A **time reference** is an infinite set T together with an internal law $+^T : T \times T \rightarrow T$ and a pointed subset $(T^+, 0^T)$ satisfying the following conditions:

- upon T^+: closure, initiality, neutral to left
- upon T: associativity, neutral to right, cancelable to left, linearity

Examples : $\mathbb{N}$, $\mathbb{R}$, ${}^*\mathbb{R}$ (set of nonstandard real numbers containing infinitesimal, normal & infinite real numbers), VHDL.

Time scales (to define systems' variables within time)

Definition (Time scale)

A **time scale** is any subset $\mathbb{T}$ of a time reference T such that:

- $\mathbb{T}$ has a minimum $m^{\mathbb{T}} \in \mathbb{T}$ such that $0 \preceq m^{\mathbb{T}}$
- $\forall t \in T,\ \mathbb{T}_{t+} = \{t' \in \mathbb{T} | t' \succ t\}$ has a minimum $succ^{\mathbb{T}}(t)$
- $\forall t \in T \mid t \succ m^{\mathbb{T}}$, the set $\mathbb{T}_{t-} = \{t' \in \mathbb{T} | t' \prec t\}$ has a maximum $pred^{\mathbb{T}}(t)$
- the axiom of induction is verified on $\mathbb{T}$.

Within $^*\mathbb{R}$, we can encompass **discrete & continuous times**!
$\rightarrow$ unified time for heterogeneous systems.

Proposition: Finite union of time scales

A finite union of time scales is still a time scale.

$\rightarrow$ unified time for integrated systems.

Datasets

Definition (Dataset)

A **dataset** is a 2-tuple $\mathcal{D} = (D, \mathcal{B})$ such that:

- D is a set containing a special blank ϵ
- $\mathcal{B} = (r, w)$ where $r : D \to D$ and $w : D \times D \to D$ verify

$$
\begin{aligned}
&r(\epsilon) = \epsilon && (R1)\\
&r\big(r(d)\big) = r(d) && (R2)\\
&r\big(w(d, d')\big) = r(d') && (R3)\\
&w\big(r(d'), d\big) = d && (W1)\\
&w\big(w(d, d'), r(d')\big) = w(d, d') && (W2)
\end{aligned}
$$

An alphabet of symbols + data behaviors within time in a "virtual" buffer.

Examples : persistent dataset, ponctual dataset.

Dataflows

Definition (Dataflow)

A **dataflow** over $(\mathcal{D},\mathbb{T})$ is a mapping $X : \mathbb{T} \rightarrow D$.

Definition (Sets of dataflows)

The set of all dataflows over $(\mathcal{D},\mathbb{T})$ is noted $\mathcal{D}^{\mathbb{T}}$. The set of all dataflows over $\mathcal{D}$ with any time scale on T is noted $\mathcal{D}^{T}$.

Definition (Projection of a dataflow on a time scale)

The **projection** $X_{\mathbb{T}_P}$ **of** X **on** $\mathbb{T}_P$ is the dataflow on $(\mathcal{D}, \mathbb{T}_P)$ induced (following the data behaviors) by X on $\mathbb{T}_P$.

Definition (Equivalence of dataflows as far as)

X and Y are **equivalent as far as** $t_0 \in T$ (noted $X \sim_{t_0} Y$) iif :
$\forall \mathbb{T} \in Ts(T),\ \forall t \in \mathbb{T} \mid t \preceq t_0,\quad X_{\mathbb{T}}(t) = Y_{\mathbb{T}}(t)$

Formalization of systems

Definition (System)

A **system** is a 7-tuple $\int = (\mathbb{T}_s, Input, Output, S, q_0, \mathcal{F}, \mathcal{Q})$ where

- $\mathbb{T}_s$ is the time scale of the system
- $Input = (In, \mathcal{I})$ and $Output = (Out, \mathcal{O})$ are respectively input and output datasets
- S is the non-empty set of states
- q_0 is the initial state of the system
- $\mathcal{F} : In \times S \times \mathbb{T}_s \to Out$ is the functional behavior
- $\mathcal{Q} : In \times S \times \mathbb{T}_s \to S$ is the states behavior.

$(Input, Output)$ is called the *signature* of $\int$.

Transfer functions

Definition (Transfer function)

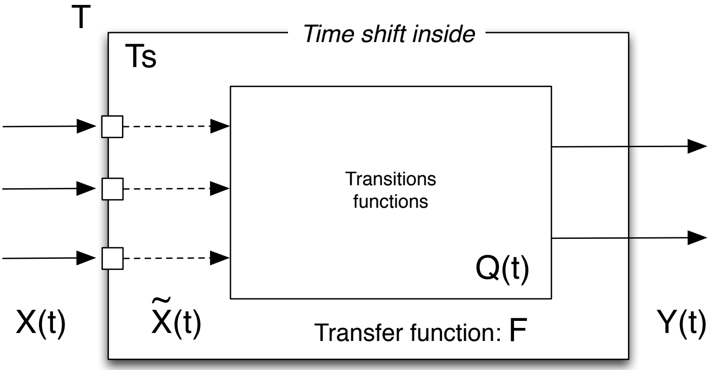
A function $F : Input^T \to Output^{\mathbb{T}_s}$ is a **transfer function** of time scale $\mathbb{T}_s$ on signature $(Input, Output)$ if, and only if:

$$\forall X, Y \in Input^T,\ \forall t \in T,\ (X_{\mathbb{T}_s} \sim_t Y_{\mathbb{T}_s}) \Rightarrow (F(X) \sim_t F(Y))$$

Theorem: Transfer function of a system

Let $\int$ be a system. There exists a unique transfer function $F_\int$, called **the transfer function of** $\int$.

Graphical representation of a system



What is integration ?

We propose three elementary operators allowing to model systems integration, i.e. to build greater systems from a set of elementary systems by recursive application of composition operators and abstraction operator.

- **Composition (divided in Product and Feedback)** consists in aggregating systems together in an overall greater system where some inputs and outputs of the various systems have been interconnected.
- **Abstraction** allows to define from a composition of systems a more abstract system that will itself be integrated in more global ones.

Extension (technical operator)

The extension of a system consists in defining it on a finer time scale (making it possible to define a finite number of systems on a shared time scale).

Definition (Extension of a system)

Let $\mathbb{T} \in Ts(T)$ be a time scale such that $\mathbb{T}_s \subseteq \mathbb{T}$. The **extension of $\int$ to $\mathbb{T}$** is the new system

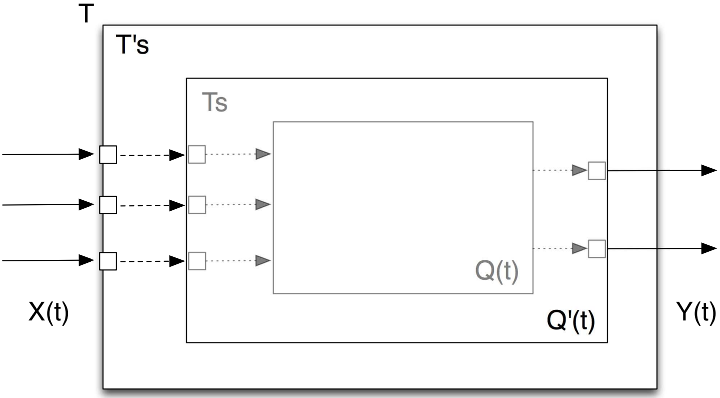
$$\int = (\mathbb{T}, Input, Output, S \times In \times Out, (q_0, \epsilon, \epsilon), \tilde{\mathcal{F}}_{\mathbb{T}}, \tilde{\mathcal{Q}}_{\mathbb{T}})^a$$

[a] $\tilde{\mathcal{F}}_{\mathbb{T}}$ and $\tilde{\mathcal{Q}}_{\mathbb{T}}$ are technical functions extending $\mathcal{F}$ and $\mathcal{Q}$.

Theorem: Equivalence of a system by extension

Let $\int$ be a system and $\int'$ be its extension to a finer time scale. Then S and S' have equivalent transfer functions: $F_{\int} \sim F_{\int'}$.

Illustration of the extension operator



Product

After defining an intuitive product on datasets and dataflows:

Definition (Product of systems)

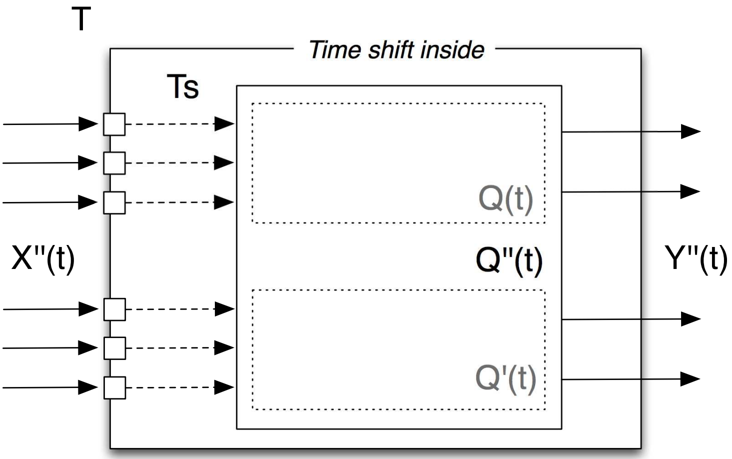
The **product** $\mathcal{S}_1 \otimes \cdots \otimes \mathcal{S}_n$ is the system

- $Input = Input_1 \otimes \cdots \otimes Input_n$ (and idem for Output)
- $S = S_1 \times \cdots \times S_n$ and $q_0 = ({q_0}_1, \ldots, {q_0}_n)$
- $\mathcal{F}\big((x_1, \ldots, x_n), (q_1, \ldots, q_n), t\big) = \big(\mathcal{F}_1(x_1, q_1, t), \ldots, \mathcal{F}_n(x_1, q_1, t)\big)$
- $\mathcal{Q}\big((x_1, \ldots, x_n), (q_1, \ldots, q_n), t\big) = \big(\mathcal{Q}_1(x_1, q_1, t), \ldots, \mathcal{Q}_n(x_1, q_1, t)\big)$

Theorem: Consistency of the product of systems

The transfer function of the product is equivalent to the “intuitive” product of the transfer functions: $F_{\int 1 \otimes \cdots \otimes \int n} \sim F_{\int 1} \otimes \cdots \otimes F_{\int n}$

Illustration of the product operator



Feedback

Definition (Feedback of a system)

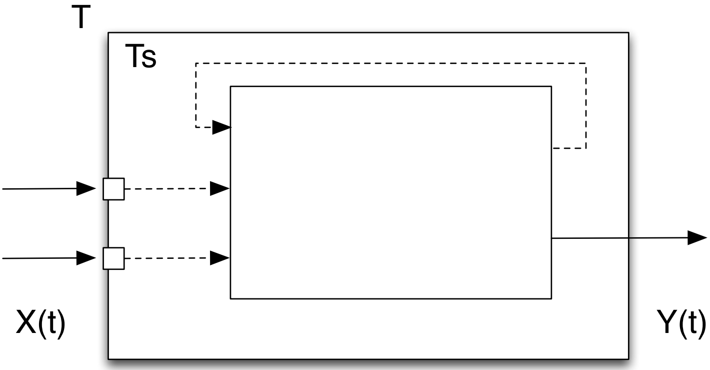
When there is no instantaneous influence of dataset D from the input to the output, the **feedback of D in $\int$** is the system $\int_{fb(D)} = \left(\mathbb{T}_s, (In, \mathcal{I}'), (Out, \mathcal{O}'), S, q_0, \mathcal{F}', \mathcal{Q}'\right)$ where

- we note $d_{x,q,t} = \mathcal{F}\left((\epsilon, x), q, t\right)_D$
- $\mathcal{I}'$ is the restriction of $\mathcal{I}$ to In
- $\mathcal{O}'$ is the restriction of $\mathcal{O}$ to Out
- $\mathcal{F}'(x \in In, q \in S, t) = \mathcal{F}\left((d_{x,q,t}, x), q, t\right)_{Out}$
- $\mathcal{Q}'(x \in In, q \in S, t) = \mathcal{Q}\left((d_{x,q,t}, x), q, t\right)$

Theorem: Consistency of the feedback on systems

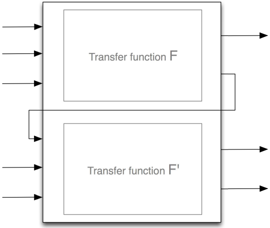
The transfer function of the feedback of a system is the "intuitive" feedback of the transfer function of this system: $F_{\int_{fb(D)}} = fb_{(F_\int, D)}$

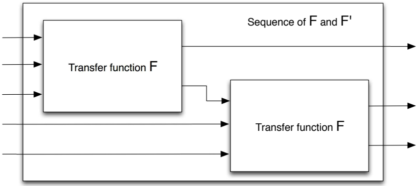
Illustration of the feedback operator



Composition from product and feedback

We have divided the composition into two operators (product and feedback). The composition of n systems can be easily obtained from product and feedback:





Abstraction

Definition (Abstraction/concretization of dataflows)

An **abstraction** of dataflows is a function $A : {D_c}^{\mathbb{T}_c} \rightarrow {D_a}^{\mathbb{T}_a}$ which is causal: $\forall X, Y \in {D_c}^{\mathbb{T}_c},\ \forall t \in T,\ (X \sim_t Y) \Rightarrow (A(X) \sim_t A(Y))$
The **concretization** associated to A is the function $C : {D_a}^{\mathbb{T}_a} \rightarrow \mathcal{P}\left({D_c}^{\mathbb{T}_c}\right)$ defined by $C(X) = A^{-1}(\{X\})$.

Definition (Abstraction of a transfer function)

The **abstraction of** F for input and output abstractions (A_i, A_o) is the new transfer function $F_a : (Input_a \otimes \mathcal{E})^T \rightarrow {Output_a}^{\mathbb{T}_a}$ defined by: $\forall X \in Input^T,\ \exists E \in \mathcal{E}^{\mathbb{T}_a},\ F_a\left(A_i(X_{\mathbb{T}_s}) \otimes E\right) = A_o\left(F(X)\right)$

Abstraction of a system

Definition (Abstraction of a system)

Let $\int = (\mathbb{T}_s, Input, Output, S, q_0, \mathcal{F}, \mathcal{Q})$ be a system.
$\int' = (\mathbb{T}_a, Input_a \otimes \mathcal{E}, Output_a, S_a, q_a0, \mathcal{F}_a, \mathcal{Q}_a)$ is an abstraction of $\int$ for input and output abstractions (A_i, A_o) iif:

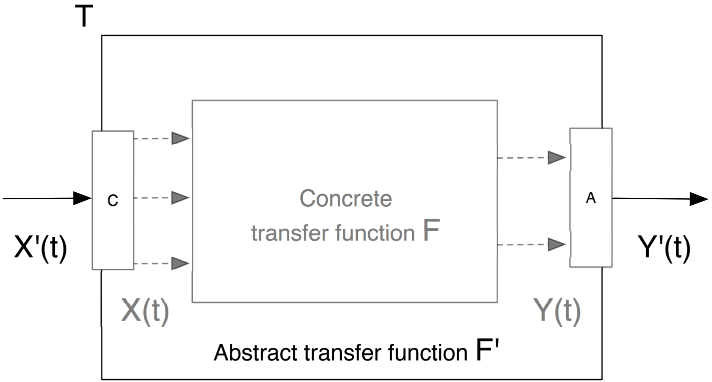
$$\exists A_q : S^{\mathbb{T}_s} \to {S_a}^{\mathbb{T}_a}, \text{ forall execution } (X, Q, Y) \text{ of } \int, \exists E \in \mathcal{E}^{\mathbb{T}_a}, \\ (A_i(X_{\mathbb{T}_s}) \otimes E, A_q(Q), A_o(Y)) \text{ is an execution of } \int'.$$

Conversely, $\int'$ is a concretization of the system $\int$.

Theorem: Consistency of the abstraction of a system

The transfer function of the abstraction of a system is the abstraction of the transfer function of this system.

Illustration of the abstraction operator



Example of a physical integrated system: a Water Tank

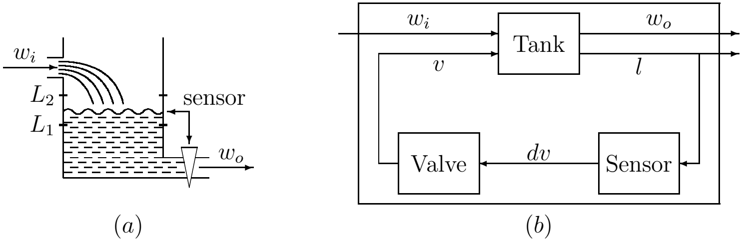


Figure 15: The Water Tank (a) and an associated simplified system (b).

Conclusion

Our "minimalist" and "unified" semantics for heterogeneous integrated systems relies on

- unified definitions of time (both continuous and discrete) and data (with various behaviors) to handle heterogeneous components
- a formal definition of systems as unified mathematical objects performing step by step transitions for changing their states and outputs.
- minimalist operators for defining systems integration with closure of the model and intuitive properties proved on transfer functions.

We believe it makes it possible to define semantics for systems approach, and later for Systems Engineering.