
Simulazione con AnyLogic

Modellazione e analisi di sistemi con il pacchetto di simulazione AnyLogic

Docente

- Luigi De Giovanni

Dipartimento di Matematica (Torre Archimede) – uff. 423

049 827 1349

luigi@math.unipd.it

www.math.unipd.it/~luigi

Ricevimento: giovedì, h 10.30 – 12.30

(**su appuntamento** via e-mail)

<https://unipd.zoom.us/j/6301138632>

Obiettivi del mini-corso

- Introdurre alla modellazione di sistemi reali con i paradigmi della *Simulazione a Eventi Discreti (Teoria della Code)* e della *System Dynamics*
- Fornire le capacità di base per modellare in modo formale sistemi reali
- Introdurre all'utilizzo di *AnyLogic*, uno strumento per la modellazione di sistemi e per la loro analisi tramite simulazione

Schema del corso

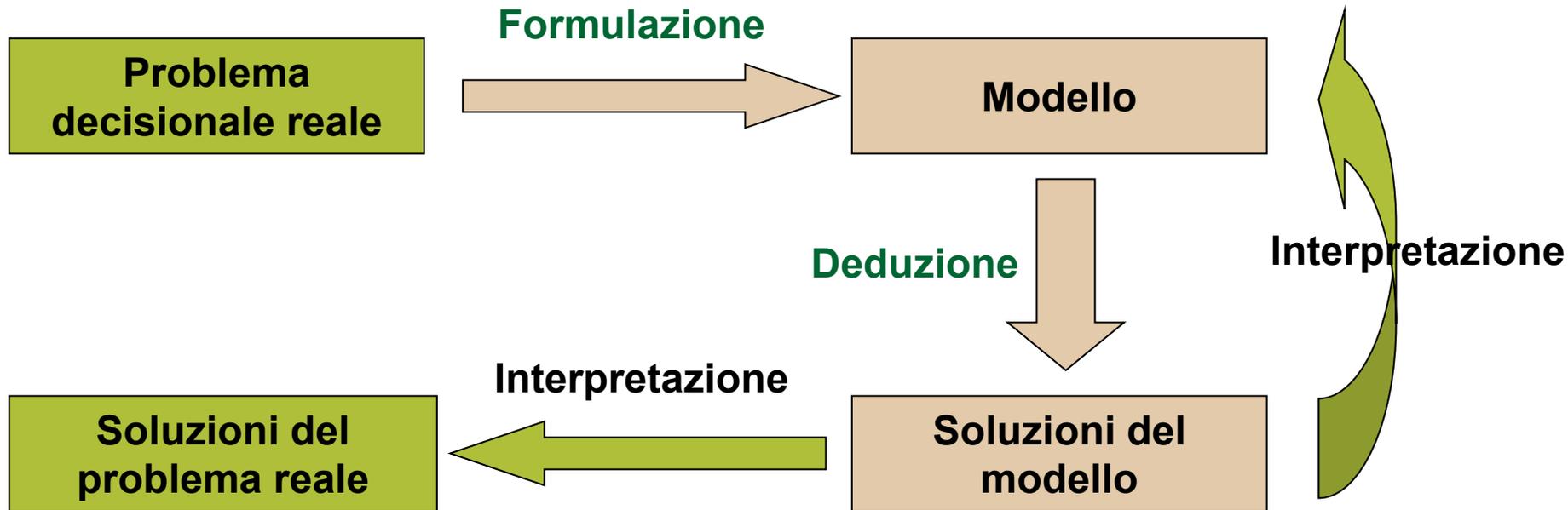
- Note introduttive
- Simulazione a eventi discreti: nozioni di base
- Simulazione a eventi discreti con AnyLogic
- Modellazione di sistemi dinamici
 - Relazioni causa-effetto e retroazioni
 - Modelli base di comportamento
- Rappresentazione di sistemi dinamici
 - Diagrammi causa-effetto
 - Reti flussi-livelli
- Simulazione di sistemi dinamici con AnyLogic
- Esempi notevoli di modellazione

Bussola

- **Note introduttive**
- Simulazione a eventi discreti: nozioni di base
- Simulazione a eventi discreti con AnyLogic
- Modellazione di sistemi dinamici
 - Relazioni causa-effetto e retroazioni
 - Modelli base di comportamento
- Rappresentazione di sistemi dinamici
 - Diagrammi causa-effetto
 - Reti flussi-livelli
- Simulazione di sistemi dinamici con AnyLogic
- Esempi notevoli di modellazione

Note introduttive

Il metodo della ricerca operativa



- **Formulazione:** modelli matematici, modelli su grafo, modelli di simulazione, modelli di teoria dei giochi etc.
- **Deduzione:** metodi quantitativi, algoritmi efficienti

La simulazione

Experience is an expensive school (Benjamin Franklin)

Experience is something you get just after you need it (!)

- Tecnica per la riproduzione (**imitazione**) del comportamento di un sistema reale in condizioni controllate
- Basata sulla costruzione di un **modello** del sistema
- Permette l'esecuzione di **esperimenti**
 - controllabili
 - manipolabili
 - riproducibili
 - che non perturbano / modificano il sistema «reale»

Scopi della simulazione

- Analisi prestazionale (valutare le caratteristiche di un sistema)
- Progetto (analisi di sistemi ipotetici)
- Confronto (tra più sistemi)
- Individuazione di punti critici (ad es. bottleneck)
- Previsione (osservare il possibile comportamento nel futuro)
- Analisi “What-if”
- Ottimizzazione (scelta dei parametri/configurazioni ideali)
- Generazione del campione di una popolazione (metodo Montecarlo)
- ...

Modelli (per la simulazione)

- **Semplificazione** della realtà: ci si limita a considerare solo alcune caratteristiche, trascurandone altre
- **Astrazione** della realtà: non si rappresenta un oggetto specifico, ma alcuni *attributi* dello stesso oggetto, in modo da poter modellare oggetti diversi di una stessa **classe** (es., non un ufficio postale, ma un «tipo» di uffici postali)
- Principio di **parsimonia**: modello più **semplice** possibile che garantisca il livello di **accuratezza** desiderato, in relazione agli **obiettivi** dello studio
- Esempio: è necessario considerare le condizioni meteorologiche o la presenza di sedie per modellare le operazioni di un ufficio postale? Dipende dagli obiettivi...

Tipi di modello

- **Modelli fisici (in scala):** architettura, galleria del vento, idraulica, costruzioni navali etc.
- **Modelli iconici (pittorici):** carte geografiche, mappe, modelli strutturali in chimica e fisica etc.
- **Modelli matematici:** calcolo di formule, bilanci, profitti e perdite etc. (si prestano ad una implementazione al calcolatore)
- **Modelli ibridi:** ad es. modelli iconici interattivi (*Visual Interactive Modelling Systems*)

Una possibile classificazione dei modelli

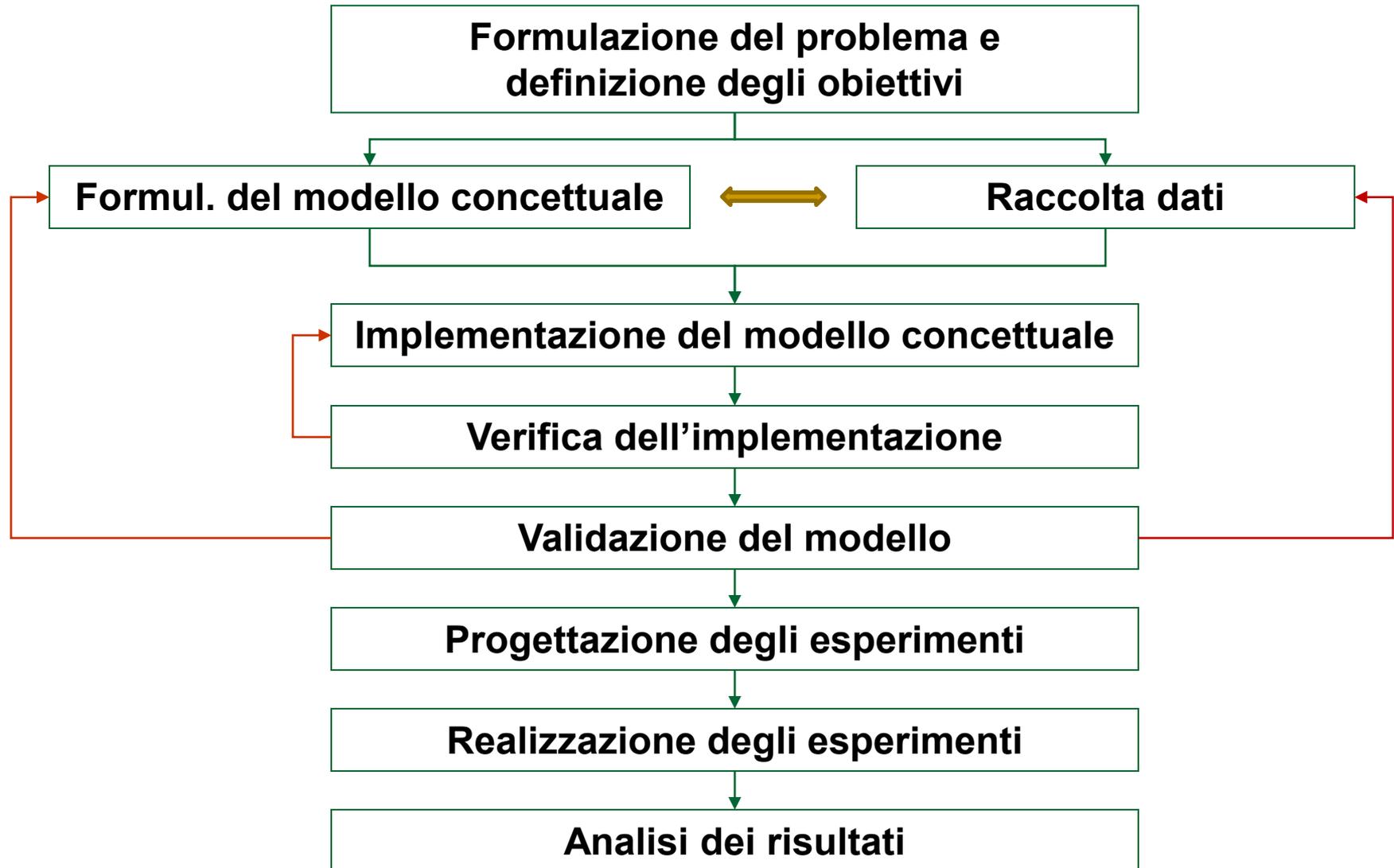


Componente temporale	Statici bilancio	Dinamici c/c bancario
Natura dei parametri	Deterministici catena di montaggio	Stocastici traffico
Tipo di evoluzione	Continui fiume	Discreti file d'attesa

Simulazione .vs. metodi analitici

- Se il modello è “sufficientemente semplice”, metodi **analitici**
 - teoria delle code
 - programmazione matematica (continua/intera)
 - equazioni differenziali (moto, diffusione, etc..)
- **Simulazione** quando:
 - il modello è molto complesso
 - non è possibile/conveniente/facile identificare le leggi matematiche che governano l'intero sistema
 - la componente stocastica è elevata

Fasi di un approccio per simulazione



Focus

- Modelli Dinamici Stocastici Discreti

Sistemi/Simulazione a Eventi Discreti (Discrete Event Simulation)

In particolare **Reti di code**

- Modelli Dinamici Stocastici Continui

Dinamica dei sistemi (System Dynamics)

Bussola

- Note introduttive
- **Simulazione a eventi discreti: nozioni di base**
- Simulazione a eventi discreti con AnyLogic
- Modellazione di sistemi dinamici
 - Relazioni causa-effetto e retroazioni
 - Modelli base di comportamento
- Rappresentazione di sistemi dinamici
 - Diagrammi causa-effetto
 - Reti flussi-livelli
- Simulazione di sistemi dinamici con AnyLogic
- Esempi notevoli di modellazione

Simulazione a eventi discreti

Simulazione a eventi discreti

- Riproduce l'evoluzione dello **stato** di un sistema per **eventi**
- Lo **stato** è definito da grandezze associate alle **entità**, in relazione agli obiettivi dello studio (lunghezza delle code, numero di clienti serviti etc.)
- Lo stato cambia solo in corrispondenza di determinati *fatti salienti*, gli **eventi** (arrivo di un nuovo cliente etc.)
- Gli eventi **cambiano lo stato** in modo istantaneo (gli eventi hanno durata trascurabile)

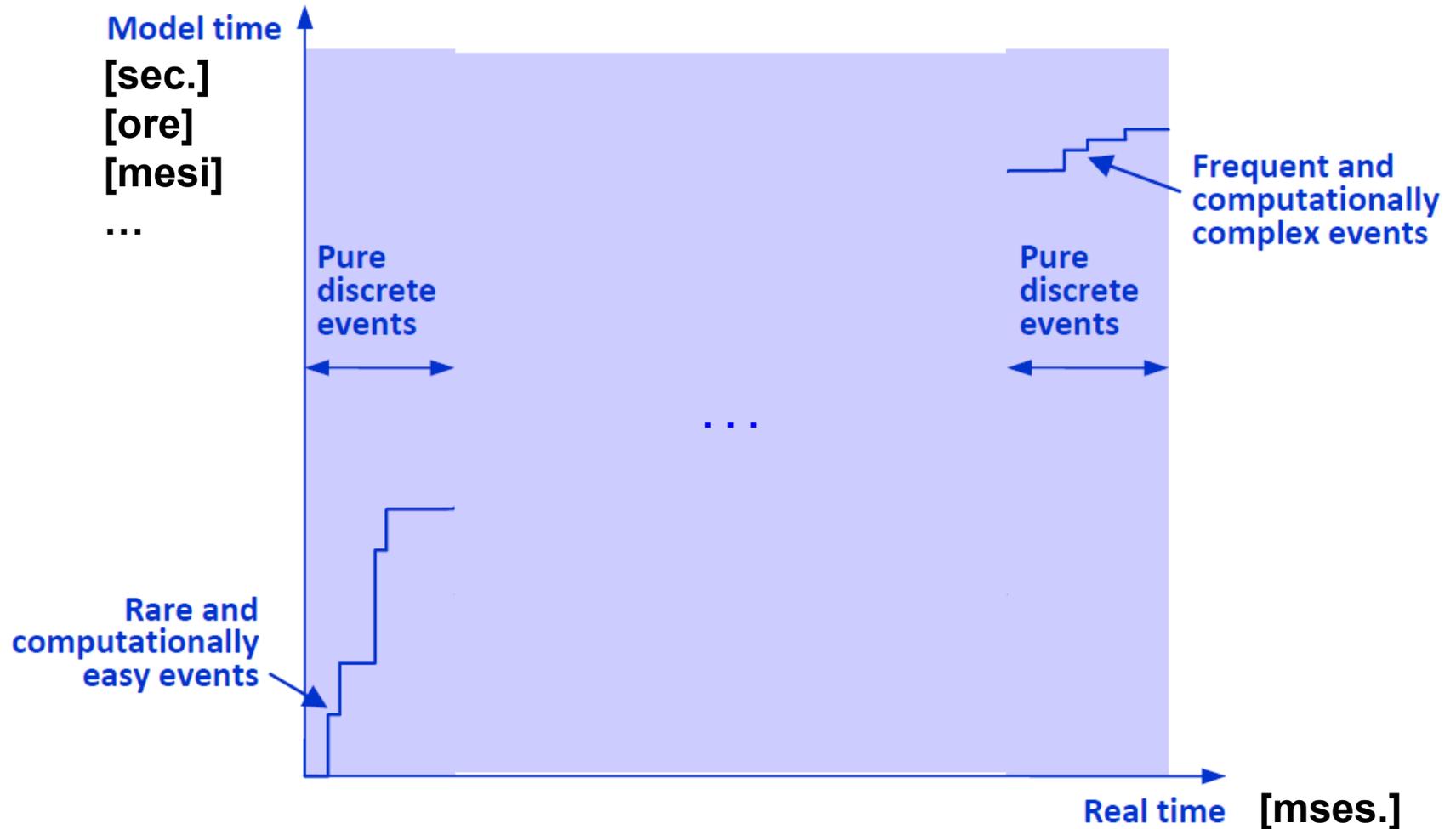
Esempio di un sistema a eventi discreti: fila d'attesa con un servente

- **Stato:** lunghezza della coda, servente libero/occupato
- **Entità:** una coda, un servente, i clienti
- **Eventi** (e ripercussioni sullo stato):
 - Evento 1 – arrivo di un nuovo cliente
 - *se il servente è occupato, la fila aumenta di una unità*
 - *se il servente è libero, il nuovo cliente occupa il servente*
 - Evento 2 – fine servizio di un cliente
 - *se la fila non è vuota:*
 - *nessun cambiamento nel servente (un nuovo cliente inizia il servizio e occupa il servente che rimane occupato)*
 - *La lunghezza della fila diminuisce di una unità*
 - *se la fila è vuota:*
 - *nessun cambiamento nella fila*
 - *il servente passa da occupato a libero*

Schema di avanzamento del tempo

1. individua e ordina temporalmente gli eventi futuri
(*es. prossimo arrivo, prossima fine servizio*)
2. fai avanzare l'orologio (*simulation [o model] time*) fino al prossimo evento (*es. arrivo o fine servizio*)
3. esegui le operazioni associate al prossimo evento, cambiando lo stato del sistema (*execution [o real] time*)
(*es. aumento o diminuzione della fila, occupazione o rilascio del servente*)
4. genera i futuri eventi, eventualmente causati da quello appena avvenuto (*lista ordinata di eventi*)
(*es. nuovo arrivo o fine del servizio che sta per iniziare*)
5. se si verificano le condizioni di *terminazione STOP*, altrimenti torna ad 1

Schema di avanzamento del tempo



© XJ Technologies www.anylogic.com

Viste del sistema

■ Vista per eventi

Focus sulla lista degli eventi, più flessibile e di basso livello, adatta per l'implementazione

■ Vista per processi

Focus sulle entità che “attraversano” il sistema, determinando/subendo azioni e decisioni. Il processo corrisponde a “percorsi” delle *entità passanti* attraverso delle *entità residenti*. Aiuta la progettazione

Bussola

- Note introduttive
- Simulazione a eventi discreti: nozioni di base
- **Simulazione a eventi discreti con AnyLogic**
- Modellazione di sistemi dinamici
 - Relazioni causa-effetto e retroazioni
 - Modelli base di comportamento
- Rappresentazione di sistemi dinamici
 - Diagrammi causa-effetto
 - Reti flussi-livelli
- Simulazione di sistemi dinamici con AnyLogic
- Esempi notevoli di modellazione

Introduzione a AnyLogic

AnyLogic

- **Ambiente di sviluppo** per l'implementazione di modelli di simulazione (un *Visual Interactive Modelling System*)
 - Interfaccia grafica per la costruzione dei modelli
 - Diversi componenti base
- Supporta **diversi paradigmi** di simulazione
 - Simulazione ad agenti (dettaglio massimo)
 - Simulazione a eventi discreti (dettaglio intermedio)
 - System Dynamics (dettaglio minimo)
 - ... e loro **ibridizzazioni** (nello stesso modello interagiscono componenti di diversi paradigmi)
- Il pieno utilizzo richiede la conoscenza di java (**non prerequisito** ai fini del mini-corso)

La guida di AnyLogic

- AnyLogic Help
 - Installazione (istruzioni in “HowToAnyLogic”)
 - Introduzione all’ambiente
 - AnyLogic functions – Probability distribution
- “Tutorials”. Importante seguire:
 - **Process Modeling Library (PML) Tutorials** (Discrete Event simulation – D.E.S.): in particolare *bank office, job shop* (attenzione: includono componenti non di interesse per il mini-corso)
 - **System Dynamics Tutorial**: *Bass diffusion*
- Library Reference Guides: descrizione sistematica dei componenti, in particolare per D.E.S. (**PML Reference Guide**)
- **AnyLogic in Three Days: A Quick Course in Simulation Modeling**
<http://www.anylogic.com/free-simulation-book-and-modeling-tutorials>
(scaricabile gratuitamente in formato pdf)
- Altro: **FAQ (HowTo)**, *Advanced modeling with Java (API etc., non trattato)*

Distribuzioni di probabilità

Per la realizzazione di distribuzioni con data media e varianza, ricordare il significato dei parametri (guida *AnyLogic Help – Stochastic Modeling – Probability distribution*)

- Esempio: funzione **gamma** (**alpha**, **beta**)
 - **alpha** (shape parameter), **beta** (scale parameter)
 - Media : **alpha · beta**
 - Varianza : **alpha · beta²**
 - *Per la distribuzione gamma di media 1 e deviazione standard 0.5, porre **alpha = 4** e **beta = 0.25***

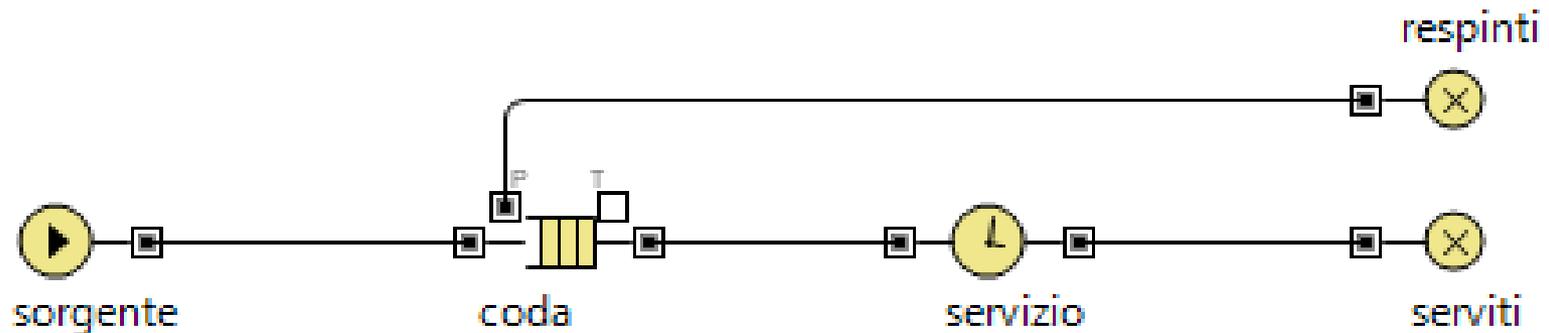
- Esempio: funzione **exponential** (**lambda**)
 - Media : **1 / lambda**

- Esempio: funzione **normal** (**sigma**, **mean**)
 - Media : **mean** Dev. std : **sigma**

Esempi di Simulazione a Eventi Discreti

Sistemi a code

- Utilizzo dei componenti della **PML**: vedremo le **componenti base** (da utilizzare in via esclusiva in questo corso)
- Esempio: coda ad un servente (`basicqueue`)



- Capacità del sistema: caso deterministico

Esercizi

1. `de.esercizi.01`

- ❑ Esempi base, code con preemption e timeout, code in serie, selettori di percorso.
- ❑ Vedi suggerimenti nel file fornito insieme ai modelli AnyLogic.

2. `de.esercizi.02`

- ❑ File d'attesa con priorità, tipi di entità personalizzate, uso di TableFunction (parametri dipendenti dal tempo), retroazioni.
- ❑ Vedi suggerimenti nel file fornito insieme ai modelli AnyLogic.

3. `de.esercizi.03`

- ❑ Percorsi articolati, esempi di applicazioni.
- ❑ Vedi suggerimenti nel file fornito insieme ai modelli AnyLogic.

Bussola

- Note introduttive
- Simulazione a eventi discreti: nozioni di base
- Simulazione a eventi discreti con AnyLogic
- **Modellazione di sistemi dinamici**
 - Relazioni causa-effetto e retroazioni
 - Modelli base di comportamento
- Rappresentazione di sistemi dinamici
 - Diagrammi causa-effetto
 - Reti flussi-livelli
- Simulazione di sistemi dinamici con AnyLogic
- Esempi notevoli di modellazione

Note Introduttive

- *System Dynamics* (Dinamica dei sistemi):
“Metodologia per analizzare i problemi in cui il **tempo** è un fattore determinante (sistemi dinamici) e che implica lo studio su come un **sistema** può essere **gestito** in relazione a **perturbazioni** esogene” (R.G. Coyle, 1977)
- Note storiche:
 - Introdotta e sviluppata negli anni '50 presso il Massachusetts Institute of Technology con il contributo fondante di J. Forrester
 - Prime applicazioni in campo sociale, con estensione negli anni '70 ad altri ambiti e, in particolare, all'economia aziendale e alla sanità pubblica

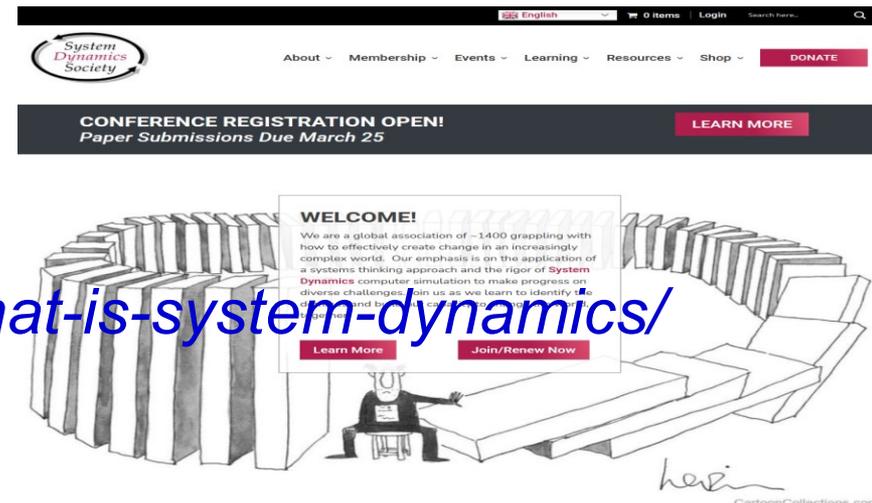
Risorse

Testi (per eventuale consultazione):

- * Sterman J., 2000, *Business Dynamics - Systems thinking and modeling for a complex world*, McGraw Hill
- Kauffman, Draper L., 1980, *Systems 1: An Introduction to Systems Thinking*, Pegasus Communications
- Roberts, 1983, *Introduction to Computer Simulation*, Portland, Oregon

Internet

- www.systemdynamics.org
- <https://systemdynamics.org/what-is-system-dynamics/>
- www.vensim.com



Modelli di System Dynamics

System Dynamics come integrazione di

- **Qualitative System Dynamics (QSD)**: metodologia generale che permette la costruzione del modello più **semplice** (ad «alto» livello, **visione sistemica**) che rappresenti il sistema allo studio, ricercando le **relazioni causa-effetto** tra elementi del sistema (interni) e tra questi e l'esterno
- **Dynamic Simulation Analysis (DSA)**: metodologia di **specificazione quantitativa** mirata alla traduzione del modello in linguaggio macchina, abilitante la **verifica** della corrispondenza tra realtà, modello e successiva **implementazione**

Perché modellare e analizzare?

Sistema: un'azienda di abbigliamento

Oggetto di interesse: livello delle vendite annuali

Situazione attuale: 80 M€

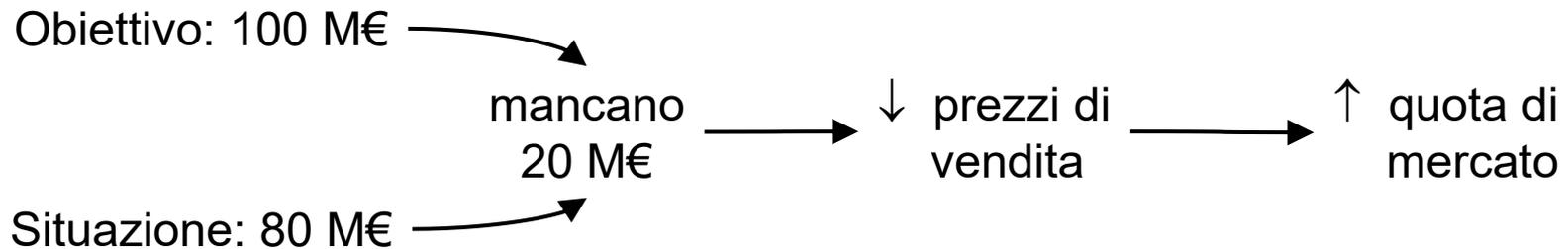
Obiettivo: 100 M€

Analisi e decisione: per aumentare le vendite di 20 M€/anno, considerando il mercato di riferimento, si possono **abbassare i prezzi** per aumentare la quota di mercato

Risultato registrato: il livello delle vendite rimane a 80 M€!!!

L'errore del management

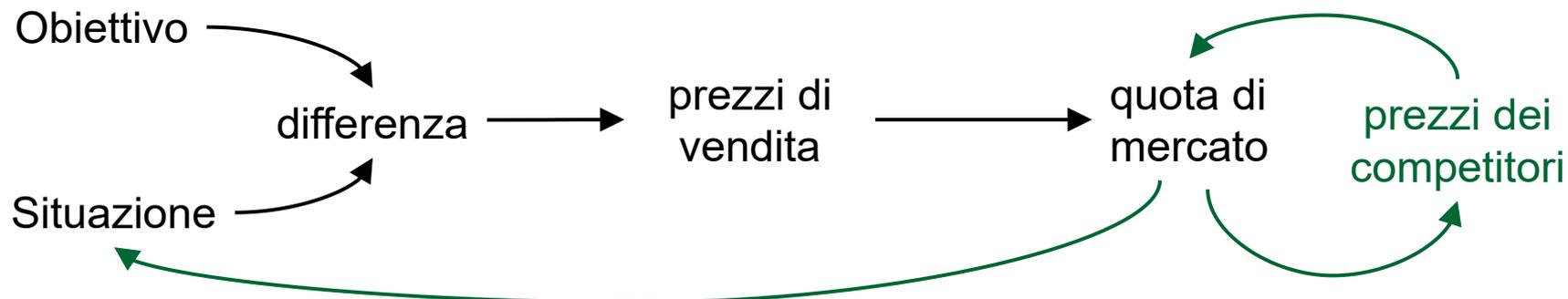
- Tendenza al ragionamento “lineare” non realistico



- Si sono trascurate le reazioni dei competitori, l'inerzia del mercato per mantenere la precedente condizione di equilibrio

Una visione più realistica

- La nostra politica dei prezzi e la reazione dei competitori influenzano la situazione attuale!!!



- La valutazione degli effetti delle nostre politiche dei prezzi diventa più difficile.

Bussola

- Note introduttive
- Simulazione a eventi discreti: nozioni di base
- Simulazione a eventi discreti con AnyLogic
- **Modellazione di sistemi dinamici**
 - **Relazioni causa-effetto e retroazioni**
 - Modelli base di comportamento
- Rappresentazione di sistemi dinamici
 - Diagrammi causa-effetto
 - Reti flussi-livelli
- Simulazione di sistemi dinamici con AnyLogic
- Esempi notevoli di modellazione

La modellazione è necessaria

I comportamenti dei sistemi sono complessi per l'ampia gamma di **interazioni** che sussistono tra le diverse componenti, più che per la complessità delle componenti stesse: **visione sistemica**

E' difficile valutare gli effetti di interventi sul sistema per:

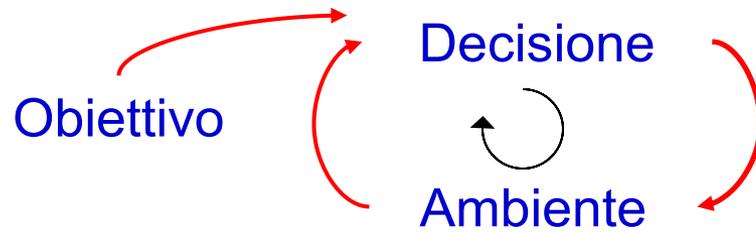
- inerzie del sistema (policy resistance)
- conseguenze non attese / desiderate (= poco studiate!)
- comportamenti contro-intuitivi

Modellare per individuare (tutte) le

- ***reazioni* del sistema (relazioni causa-effetto) e le**
- ***retroazioni* (feedback)**

Feedback

In genere, ogni azione genera una reazione che si riflette sull'azione stessa come **retroazione (feedback)**



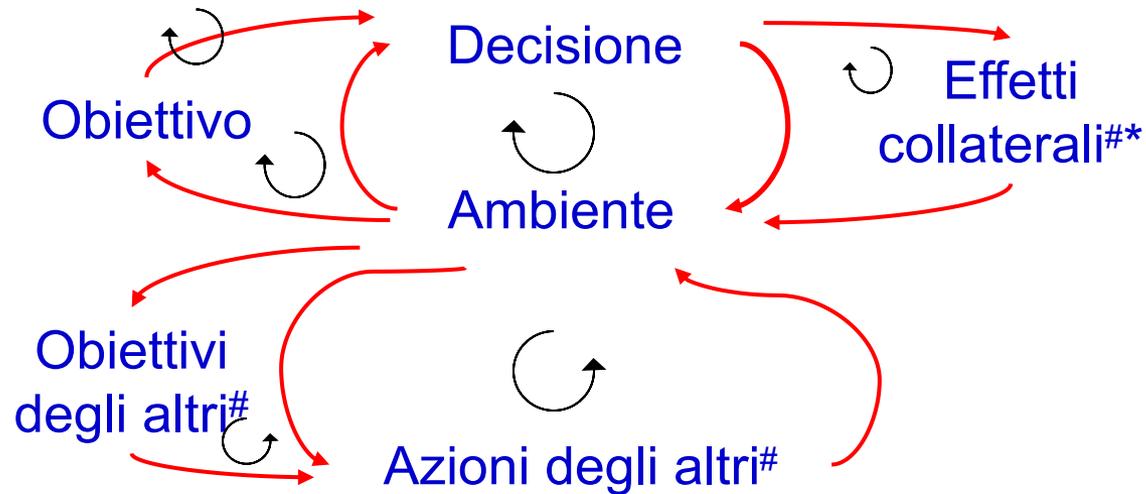
Notazione:



La retroazione è determinata da un ciclo nella catena causale (catena delle relazioni causa-effetto)

Feedback++

E' necessario approfondire l'analisi delle possibili reazioni e relative retroazioni



* Collaterali = poco studiati!

Definizione "corretta" dei confini del sistema

Relazioni causa effetto

- Relazione causa-effetto “positiva” (nello stesso senso)
 - un aumento dei prezzi di vendita dei concorrenti *tende* a far aumentare la nostra quota di mercato; una diminuzione *tende* a farla diminuire

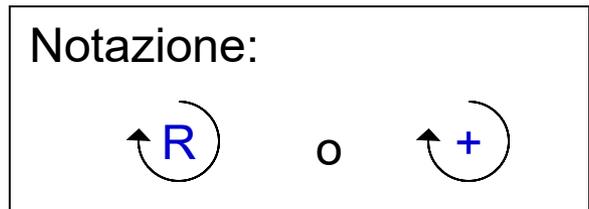
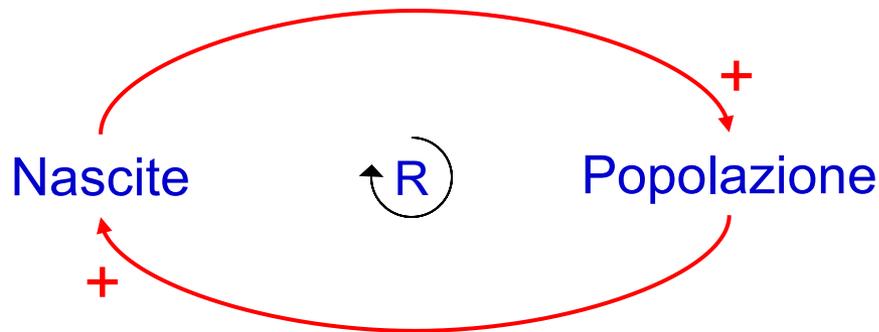
Prezzi di vendita
dei concorrenti $\xrightarrow{+}$ Nostra quota
di mercato

- ... o “negativa” (nel senso opposto):
 - un aumento dei nostri prezzi di vendita *tende* a far diminuire la nostra quota di mercato; una diminuzione *tende* a farla aumentare

Nostri prezzi
di vendita $\xrightarrow{-}$ Nostra quota
di mercato

Tipi di retroazione

- Una retroazione può essere di **Rinforzo** ...

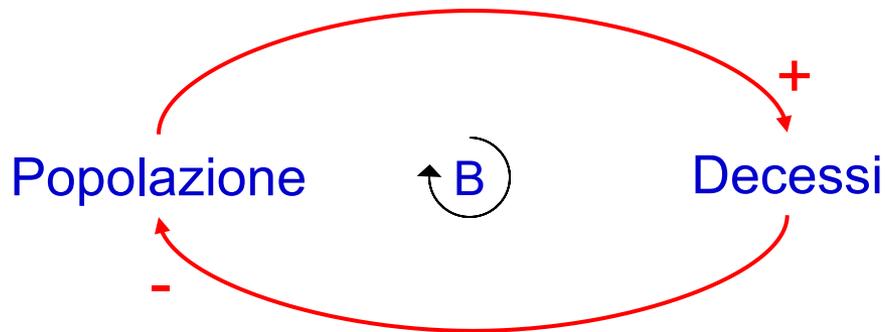


“un aumento delle nascite spinge l’aumento della popolazione, e questo aumento spinge le nascite”

- Rappresenta l’evoluzione di un processo e quindi modella una **spinta all’accelerazione** del sistema

Tipi di retroazione

- ... o di **Bilanciamento**:



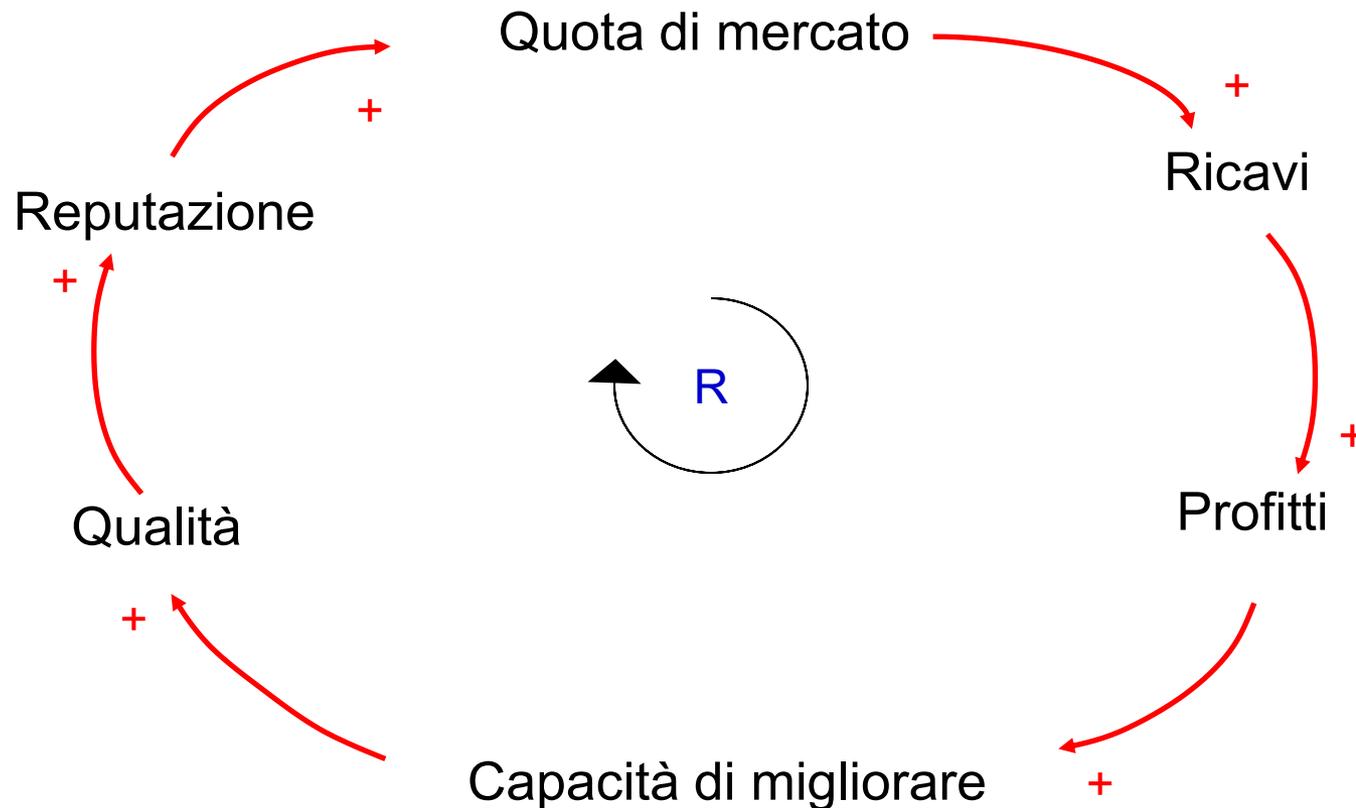
Notazione:



“un aumento dei decessi frena popolazione e questa diminuzione frena i decessi”

- Rappresenta un freno al processo e quindi modella una **spinta al rallentamento** del sistema

Esempi di retroazione



Osservazioni:

- notare la natura qualitativa del modello (ad es. non abbiamo misure della «reputazione»)
- la retroazione «rinforza» la tendenza innescata dalla variazione di un elemento nella catena

Regola d'oro della modellazione SD

- Ogni sistema dinamico, indipendentemente dalla sua complessità, può essere visto come **insieme di retroazioni** di rinforzo (positive) e di bilanciamento (negative)
 - La dinamica (il comportamento) del sistema dinamico è determinata dalla **interazione delle retroazioni**
- ⇒ **Modellare** un sistema dinamico significa individuare i **fattori** di interesse, le loro relazioni **causa-effetto** e le relative **retroazioni**

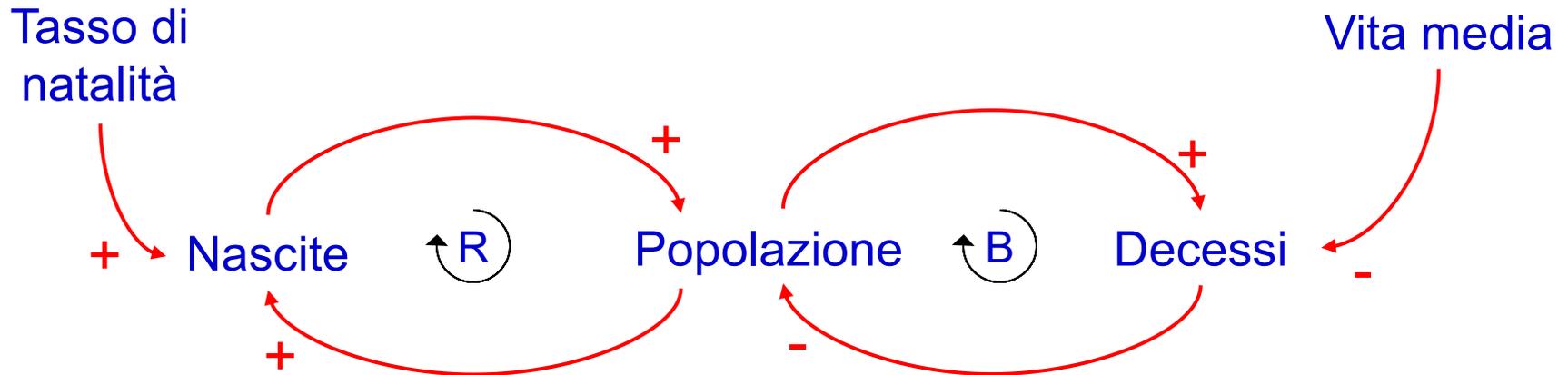
Esempio: descrizione del sistema

L'andamento della popolazione dipende dal tasso di natalità (numero di nascite nell'unità di tempo) e dall'aspettativa di vita (durata di vita media). Si vuole modellare la dinamica della popolazione.

Esempio: fattori rilevanti

- Dal testo:
 - L'andamento della popolazione dipende dal **tasso di natalità** (numero di nascite nell'unità di tempo) e dall'**aspettativa di vita** (durata di vita media). Si vuole modellare la dinamica della **popolazione**.
- Altri fattori rilevanti: la dinamica della popolazione dipende da
 - **Nascite**
 - **Decessi**

Esempio: relazioni causa effetto e retroazioni



«La dinamica della popolazione è determinata dall'interazione della retroazione di rinforzo relativa alla riproduzione e dalla retroazione di bilanciamento relativa al decesso degli individui, retroazioni governate rispettivamente dai tassi di nascita e dall'aspettativa di vita»

Modello applicabile a contesti diversi, ad esempio:

tasso natalità ~ tasso arrivi
vita media ~ durata del servizio
popolazione ~ utenti nel sistema

Descrizione di una coda ad **alto livello**

Fasi principali della modellazione SD

- Definire il **confine** del sistema
 - Individuazione dei **fattori di interesse**
 - Formulazione delle ipotesi sulla dinamica del sistema (**Dynamic Hypothesis**), ossia ipotesi sulle relazioni causa-effetto (dirette, tra coppie)
 - Determinazione delle **retroazioni** e del loro “segno”
 - **Verifica e validazione*** del sistema con eventuale ritorno alle fasi precedenti
- * in un primo momento *qualitative*: e.g., è vero che una grandezza in un feedback di bilanciamento tende a frenare la crescita di un'altra grandezza nello stesso feedback?)

Bussola

- Note introduttive
- Simulazione a eventi discreti: nozioni di base
- Simulazione a eventi discreti con AnyLogic
- **Modellazione di sistemi dinamici**
 - Relazioni causa-effetto e retroazioni
 - **Modelli base di comportamento**
- Rappresentazione di sistemi dinamici
 - Diagrammi causa-effetto
 - Reti flussi-livelli
- Simulazione di sistemi dinamici con AnyLogic
- Esempi notevoli di modellazione

Modelli base di comportamento

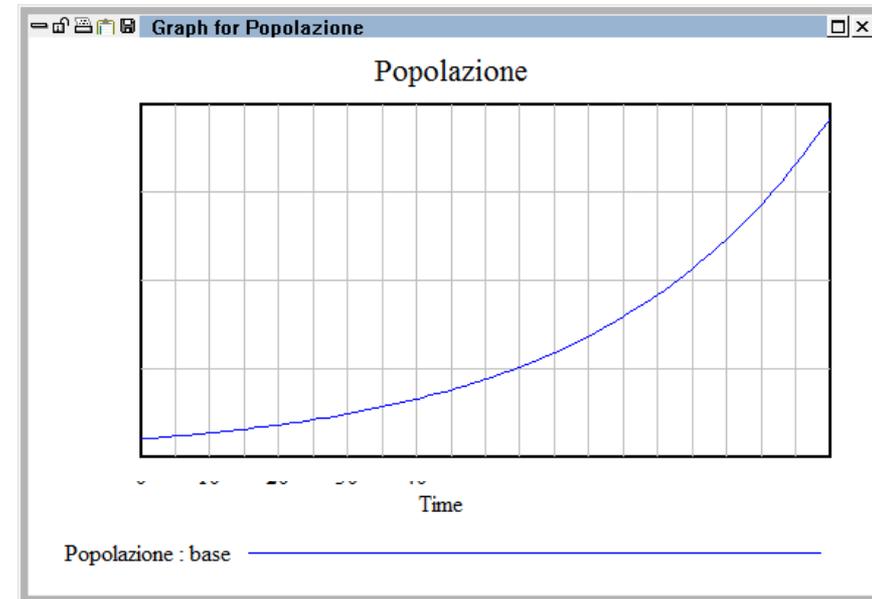
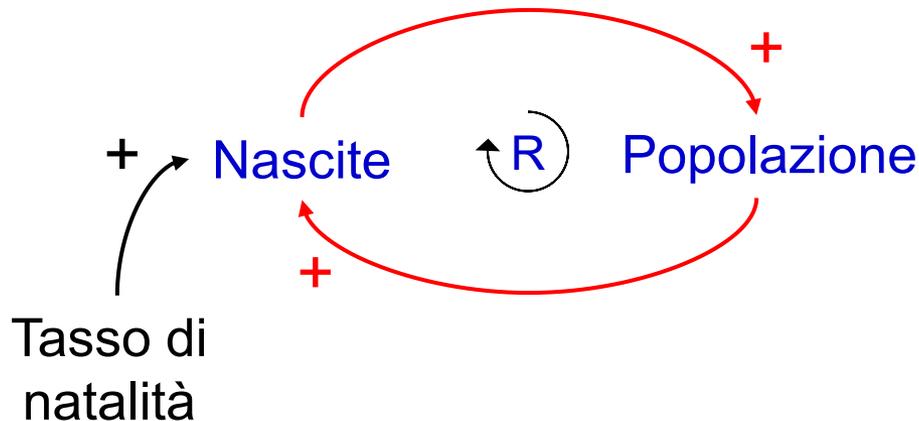
I modelli di comportamento dei sistemi dinamici possono essere ottenuti come combinazione delle seguenti **dinamiche di base**:

- Crescita o decadimento esponenziale
- Goal seeking
- Oscillazione

Ciascuna è determinata da una specifica *struttura di retroazione*

Crescita esponenziale

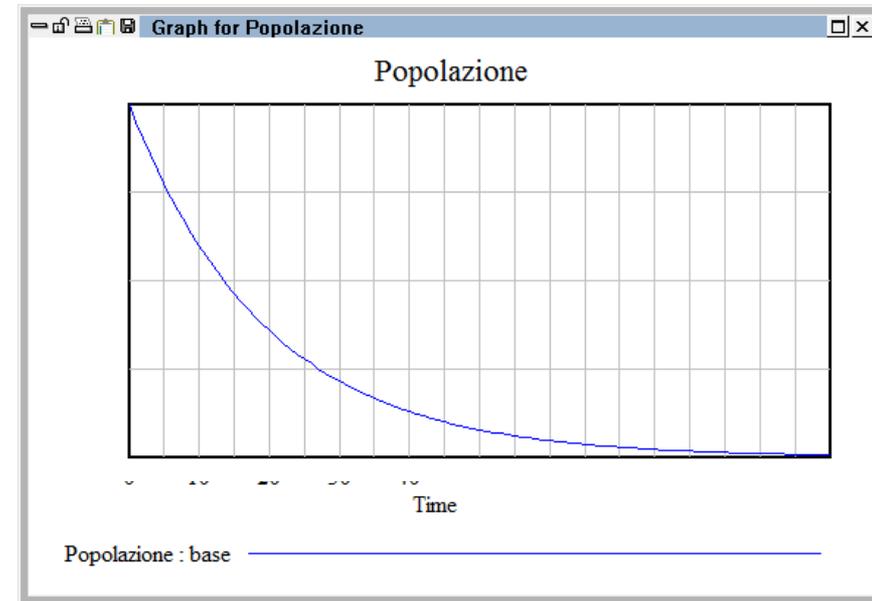
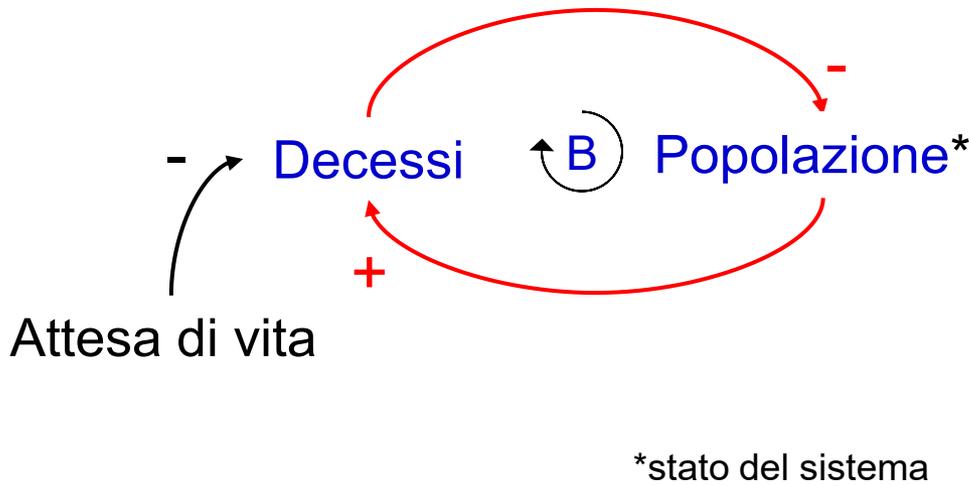
- Origina da un **feedback di rinforzo**



- Proprietà: il tempo di raddoppio dello stato del sistema è costante, indipendentemente dallo stato di partenza (=andamento esponenziale)
- Attenzione: “tutto ciò che cresce è limitato...”

Decadimento esponenziale

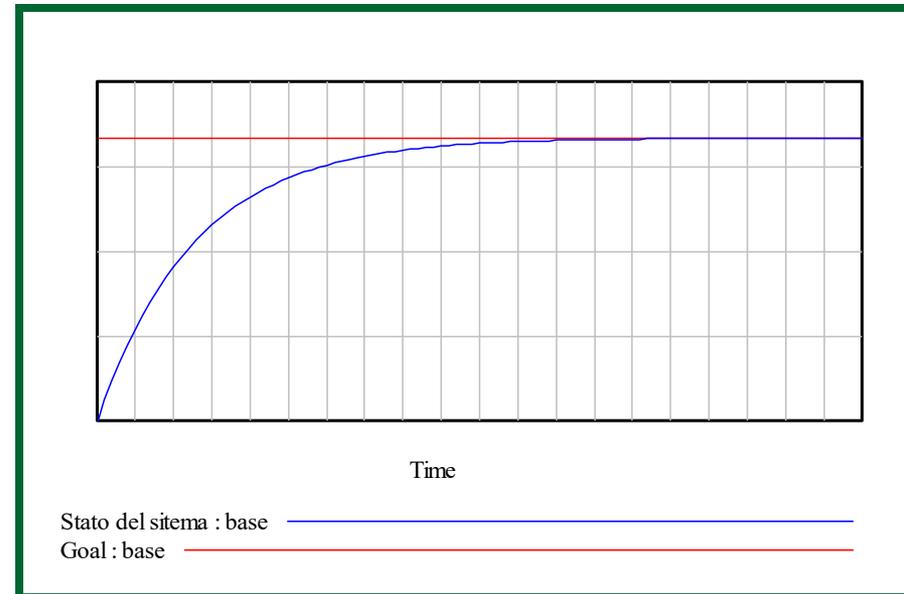
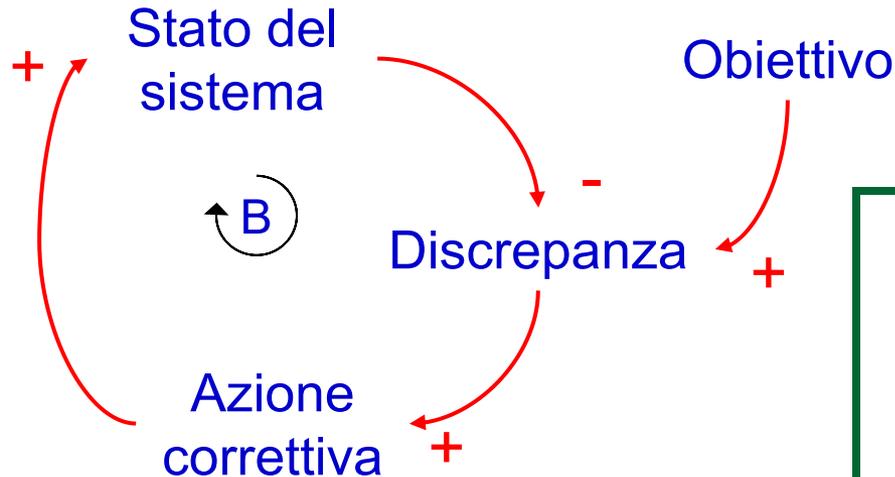
- Origina da un **feedback di bilanciamento**



- Proprietà: il tempo di dimezzamento dello stato del sistema è costante, indipendentemente dallo stato di partenza (=andamento esponenziale)

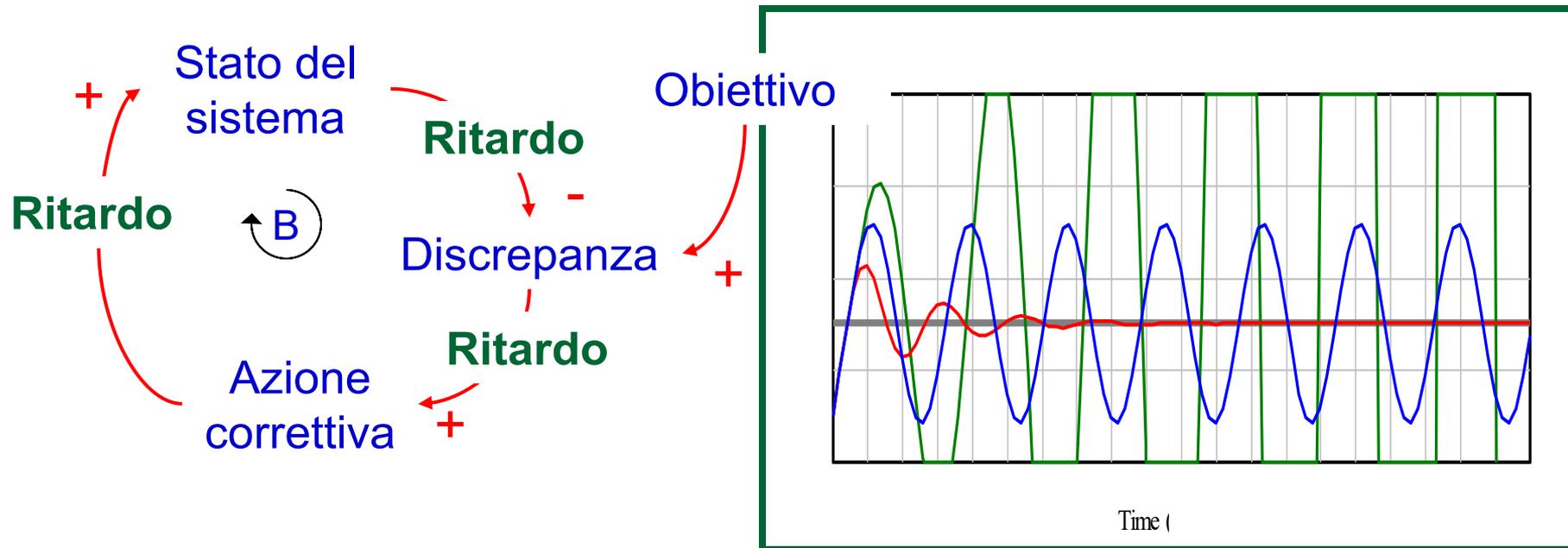
Goal Seeking

- Origina da un **feedback negativo** che vuole riportare lo stato del sistema ad un **obiettivo** prefissato, contrastando le azioni che lo allontanano



Oscillazione

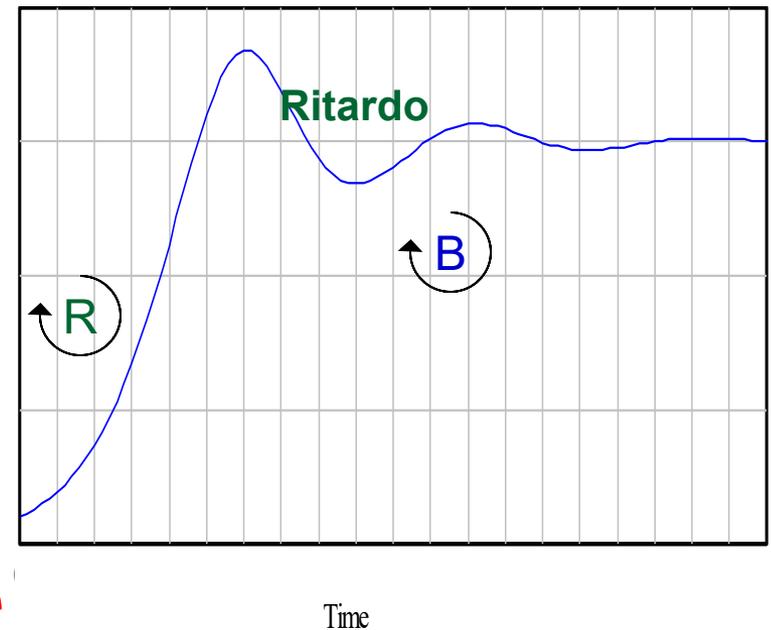
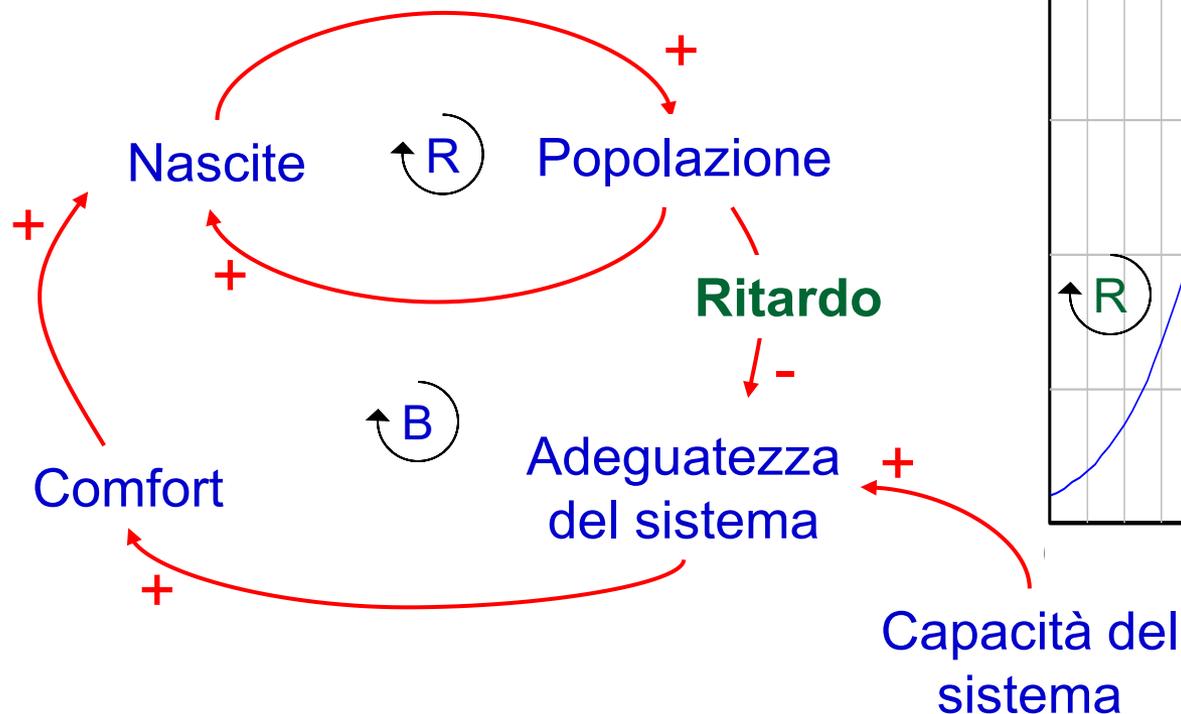
- Origina da un feedback negativo **con ritardo**



- Oscillazione provocata da un ritardo di tempo significativo che causa un'azione correttiva anche dopo che il sistema ha raggiunto l'obiettivo, forzando il sistema oltre l'obiettivo stesso (nei due sensi), e generando una reazione opposta.

Modelli di comportamento complessi

- I modelli di comportamento complessi sono il risultato dell'interazione dei modelli di base
- *Esempio:* dinamiche di crescita con limiti di capacità del sistema (crescita a “esse”) e ritardi (oscillazioni)



Modellazione di comportamenti noti

Dato un comportamento, una guida per la la comprensione della dinamica di sistema e la modellazione può essere l'individuazione delle dinamiche di base che la compongono (\approx reverse engineering)

Nell'esempio, si può intuire la presenza di un ciclo R e di un ciclo B (iniziale crescita a "esse") e di un ulteriore ciclo B (successivo decadimento esponenziale) con dei ritardi (fenomeni oscillatori)



Suggerimenti pratici

- Costruire un modello per risolvere un **particolare problema**, non «modellare il sistema»: escludere i fattori non rilevanti per assicurare che lo scopo del progetto sia fattibile e i risultati tempestivi (principio di parsimonia)
- Chiedersi se il modello dinamico è la giusta tecnica per affrontare il/i problema/i: per lo stesso sistema possono essere opportuni modelli diversi (eventualmente da integrare), a seconda del problema, degli obiettivi dello studio, del livello di dettaglio desiderato etc. (e.g., ufficio smistamento pratiche: code giornaliere .vs. annuali, dettaglio dei dati da considerare etc.)
- La simulazione dinamica non è un sostituto di altri strumenti di analisi, ma un **complemento** (e.g., individuo relazioni con SD, quantifico con DE)
- **Validazione** passo passo del modello: confrontare i risultati con i dati reali!
- **Collaborazione con il committente**: conviene che segua la definizione del modello e intervenga nella validazione nelle diverse fasi, per verificare che risponda ai quesiti richiesti e fornire nuovi suggerimenti. La notazione usata per i modelli è intuitiva e stimola la collaborazione.
- Seguire **sviluppo a spirale**: iniziare con una struttura semplice del modello, che deve essere testata, e solo successivamente entrare nel dettaglio

Bussola

- Note introduttive
- Simulazione a eventi discreti: nozioni di base
- Simulazione a eventi discreti con AnyLogic
- Modellazione di sistemi dinamici
 - Relazioni causa-effetto e retroazioni
 - Modelli base di comportamento
- **Rappresentazione di sistemi dinamici**
 - Diagrammi causa-effetto
 - Reti flussi-livelli
- Simulazione di sistemi dinamici con AnyLogic
- Esempi notevoli di modellazione

Formalizzazione di modelli SD

Esistono due tipi di rappresentazione formale

- **Diagrammi causa-effetto**

(Causal Loop Diagrams – CLD)

- permettono di cogliere la **struttura** del sistema (aspetto **qualitativo - QSD**)

- **Diagrammi o reti livello-flusso**

(Stock-Flow Networks – SFN)

- Permettono di cogliere la componente **dinamica** del sistema (aspetto **quantitativo - DSA**)

Bussola

- Note introduttive
- Simulazione a eventi discreti: nozioni di base
- Simulazione a eventi discreti con AnyLogic
- Modellazione di sistemi dinamici
 - Relazioni causa-effetto e retroazioni
 - Modelli base di comportamento
- **Rappresentazione di sistemi dinamici**
 - **Diagrammi causa-effetto**
 - Reti flussi-livelli
- Simulazione di sistemi dinamici con AnyLogic
- Esempi notevoli di modellazione

Diagrammi causa-effetto

Il formalismo permette di:

- identificare gli elementi del sistema: **variabili**
- individuare le **interdipendenze** tra le variabili (relazioni causali)
- riconoscere le retroazioni (**feedback loops**)

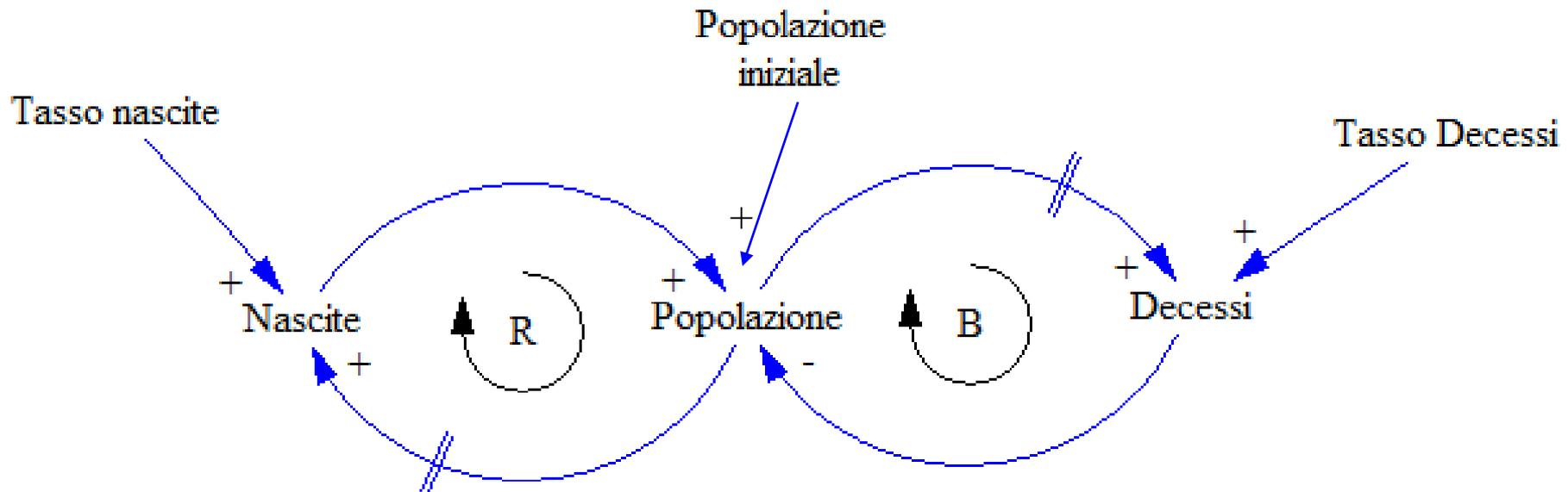
Favorisce una **descrizione qualitativa** di

- **struttura del sistema** (elementi e relazioni)
- **possibili tendenze nelle dinamiche** delle grandezze coinvolte (da modelli base comportamento)

Formalismo adatto ad analisi descrittive «preliminari»

Formalismo dei CLD

Nome	↔	Variabile
Freccia	↔	Relazione causa-effetto
Ciclo R / B	↔	Feedback di rinforzo / bilanciamento
//	↔	Ritardo



Polarità dei vettori causali

Indica il tipo di relazione causale:

- **Positiva** [+ o S = stessa direzione] se l'incremento (risp. decremento) della variabile-causa implica un incremento (risp. decremento) della variabile-effetto in misura superiore a quella che altrimenti avrebbe avuto(*): **spinta**
- **Negativa** [- o O = opposta direzione] se l'incremento (risp. decremento) della variabile-causa implica un decremento (risp. incremento) della variabile-effetto in misura superiore a quello che altrimenti avrebbe avuto: **freno**

* *Esempio: se il tasso di nascita diminuisce, la popolazione continua a crescere, ma meno di quanto faceva prima della diminuzione.*

Esempi

Qualità del prodotto $\xrightarrow{+}$ Vendite

Nascite \xrightarrow{S} Popolazione

Prezzo del prodotto $\xrightarrow{-}$ Vendite

Decessi \xrightarrow{O} Popolazione

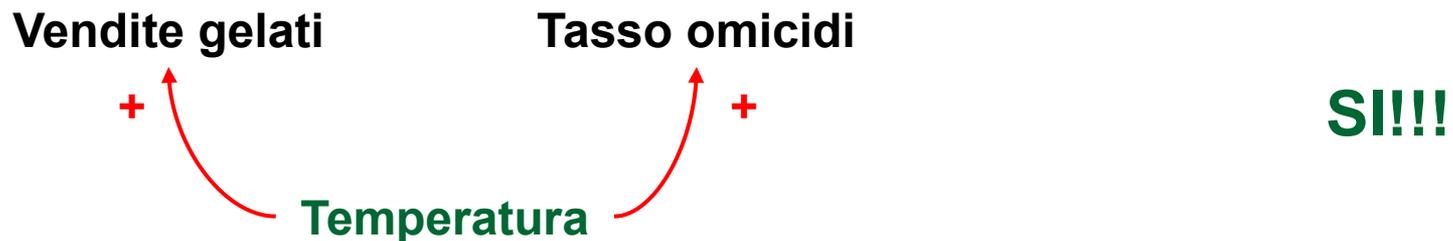


Contatti tra persone $\xrightarrow{+}$ // $\xrightarrow{+}$ Persone infette

Ritardo: tempo tra l'attivazione/variazione di una variabile-causa e l'osservazione dell'effetto. Nell'esempio un periodo di incubazione non trascurabile. Altri esempi: tempo necessario per identificare la criticità di una situazione, tempo necessario per prendere una decisione, tempo intercorrente tra la decisione e l'osservazione di un effetto.

Causalità e correlazione

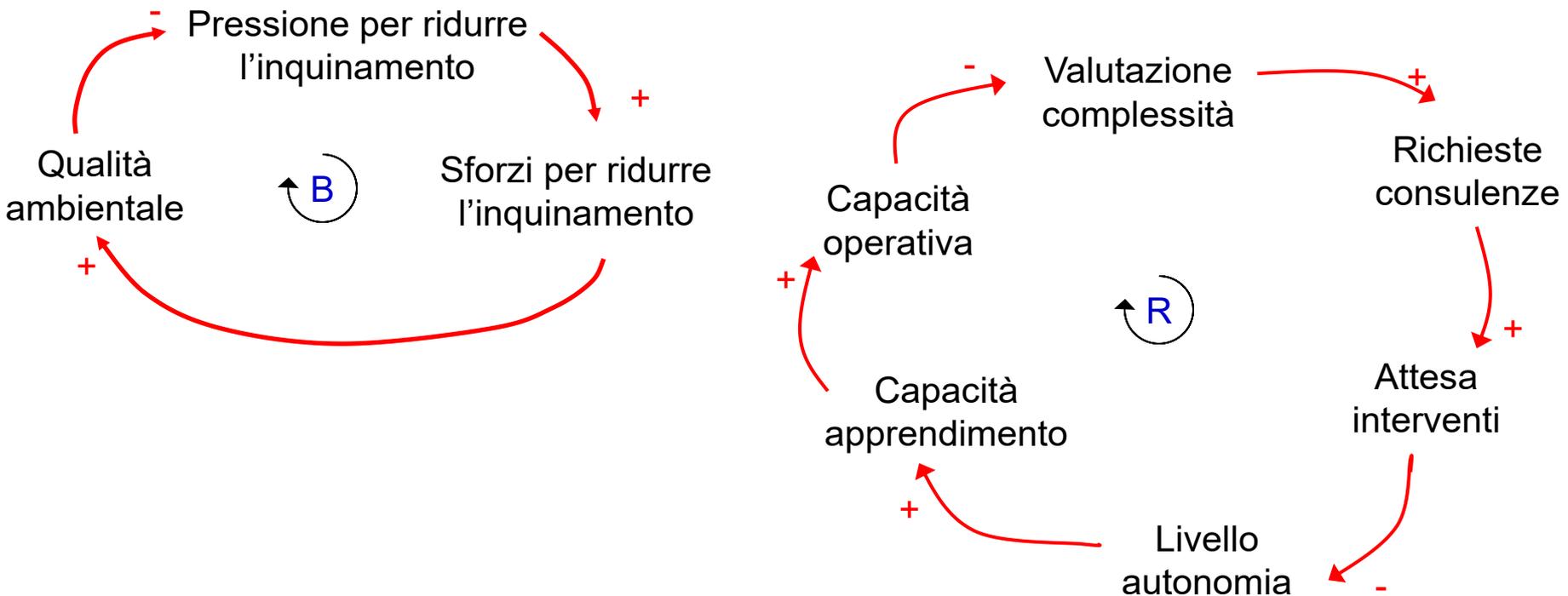
- **Nesso causale \neq correlazione**
- Le correlazioni tra variabili riflettono il comportamento passato del sistema e quindi **non** rappresentano la struttura del sistema
- La correlazione può nascondere delle relazioni causa effetto tramite terze variabili. La correlazione dovrebbe **emergere** dall'analisi del modello (o dalla sua simulazione) [e validare il modello]
- Esempio: correlazione osservata tra vendite di gelati e tasso omicidi



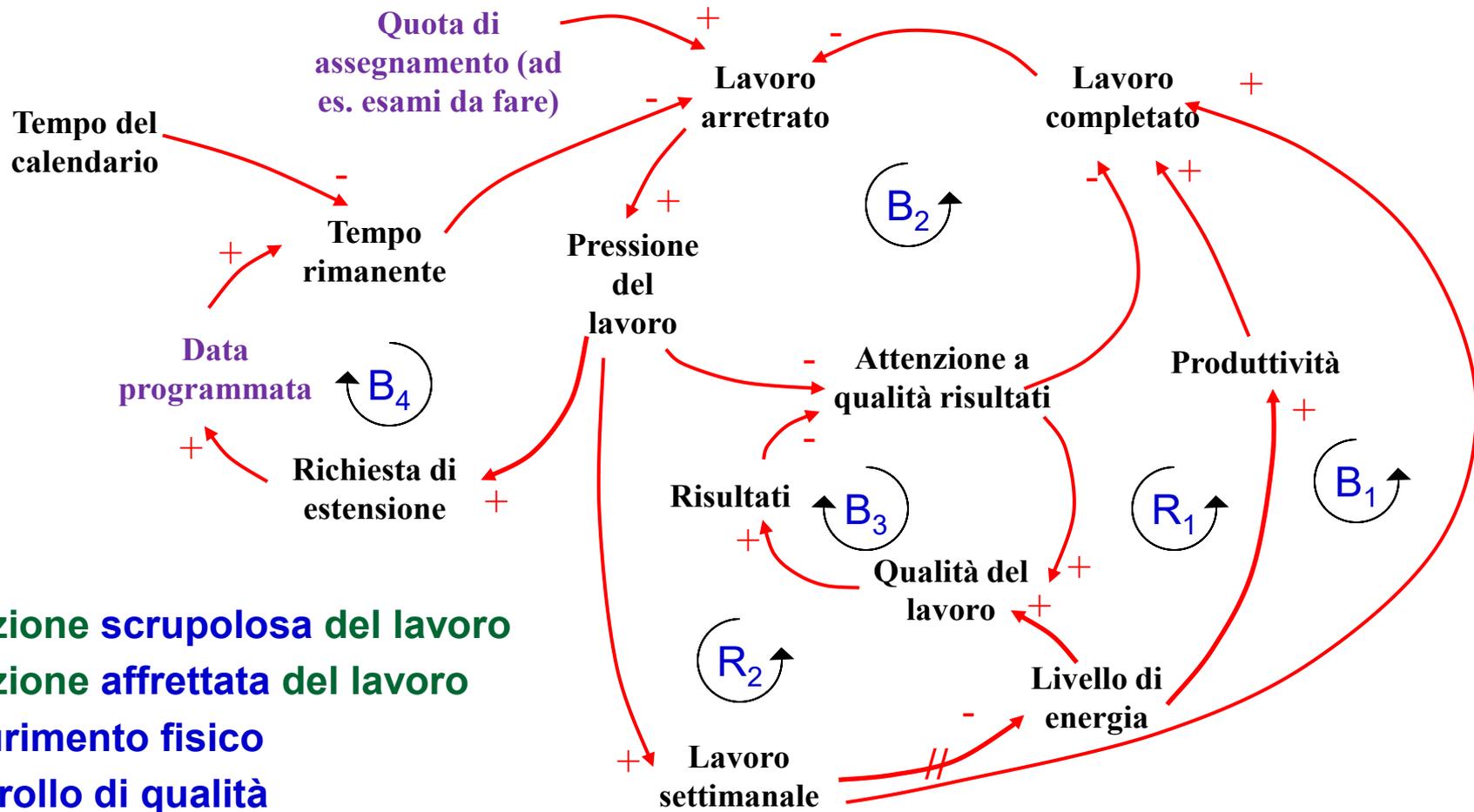
Polarità dei feedback

Per determinare se un feedback è di rinforzo o di bilanciamento, percorrere il circuito contando quanti siano i **segni negativi** sugli archi:

- se il numero è **pari** (o zero), il circuito è **di rinforzo**
- se il numero è **dispari**, il circuito è **di bilanciamento**



Un esempio: carico di lavoro



- B₁**: riduzione scrupolosa del lavoro
- B₂**: riduzione affrettata del lavoro
- R₁**: esaurimento fisico
- B₃**: controllo di qualità
- R₂**: ansia, tilt (esaurimento mentale)
- B₄**: prendere tempo

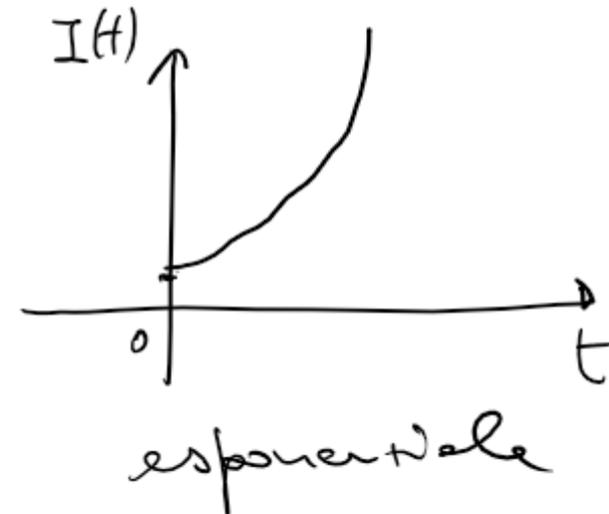
Alcune raccomandazioni pratiche

- Dare sempre una polarità alle relazioni causa-effetto
- Identificare TUTTI i feedback nel diagramma, interpretarli e validarli, dando un nome
- Esplicitare il più possibile gli obiettivi dei feedback di bilanciamento
- Indicare i ritardi di rilievo
- Scegliere nomi (non verbi) per le variabili, preferendo il nome che esprime il senso positivo (facilita la lettura delle relazioni causa-effetto e dei feedback)

Esempio: contagi (i)

Descrivere il possibile andamento delle infezioni $I(t)$

- Modello di avvio «I» (infetti)

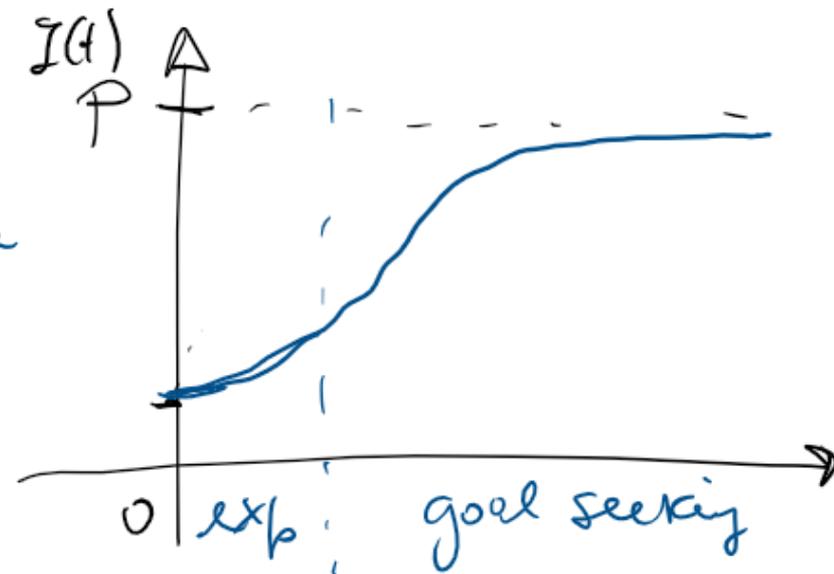
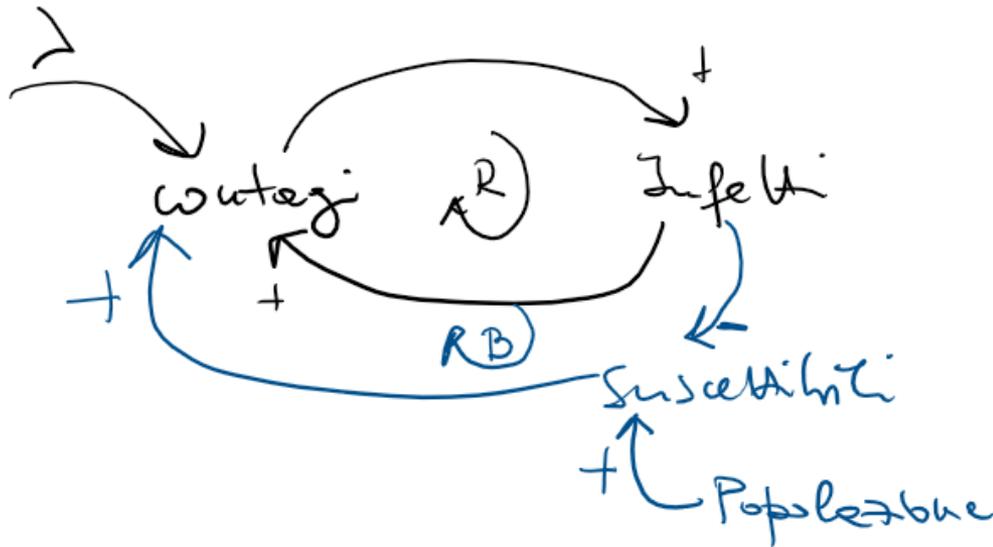


Il fenomeno dei contagi, governati dal livello di contagiosità della malattia (tasso λ), determina gli infetti

Esempio: contagi (ii)

Descrivere il possibile andamento delle infezioni $I(t)$

- Modello «S-I» (suscettibili - infetti)

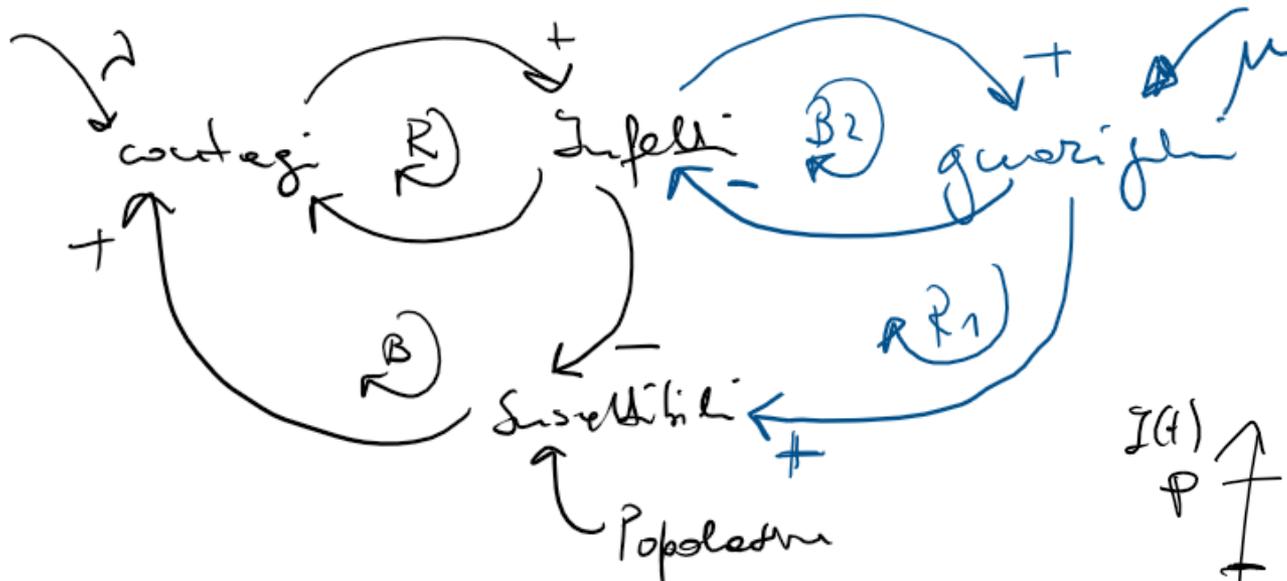


Gli infetti, attraverso il fenomeno dei contagi, portano altri individui della popolazione a diventare infetti

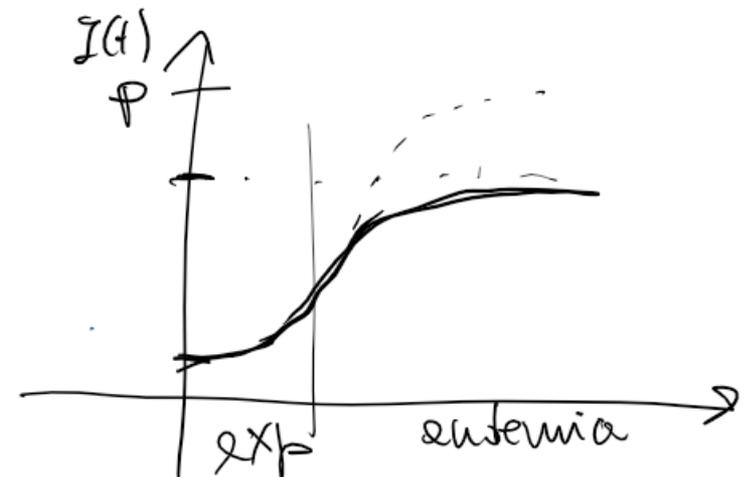
Esempio: contagi (iii)

Descrivere il possibile andamento delle infezioni $I(t)$

- Modello «S-I-S» (suscettibili – infetti - suscettibili)



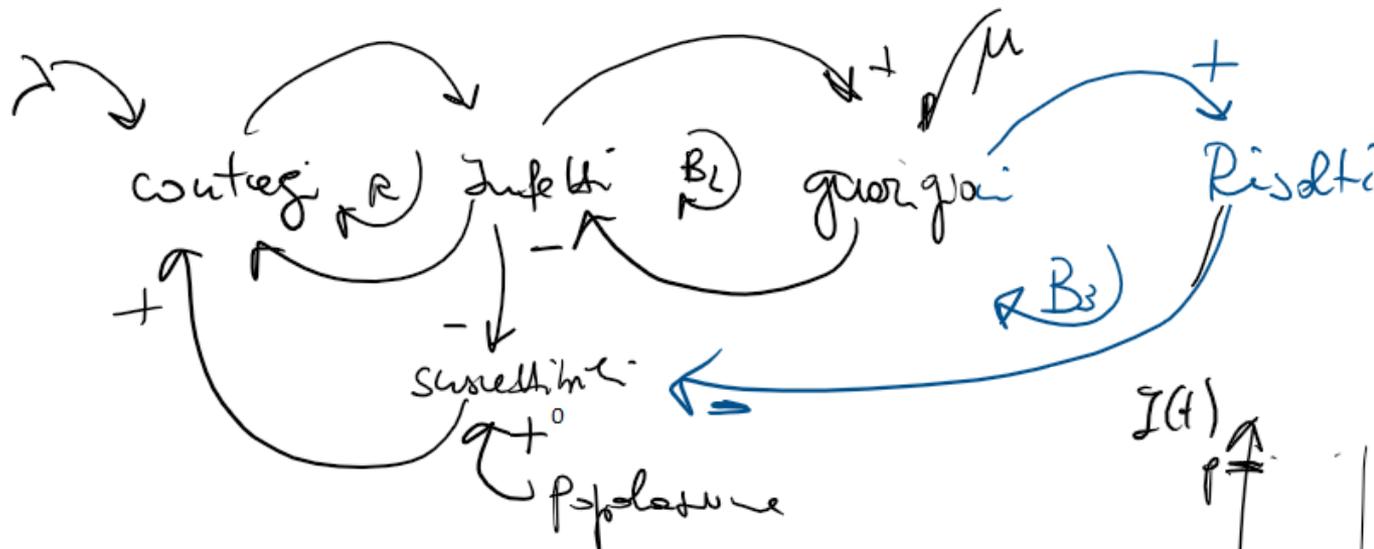
... gli infetti, con un tasso μ , guariscono ma restano suscettibili alla malattia



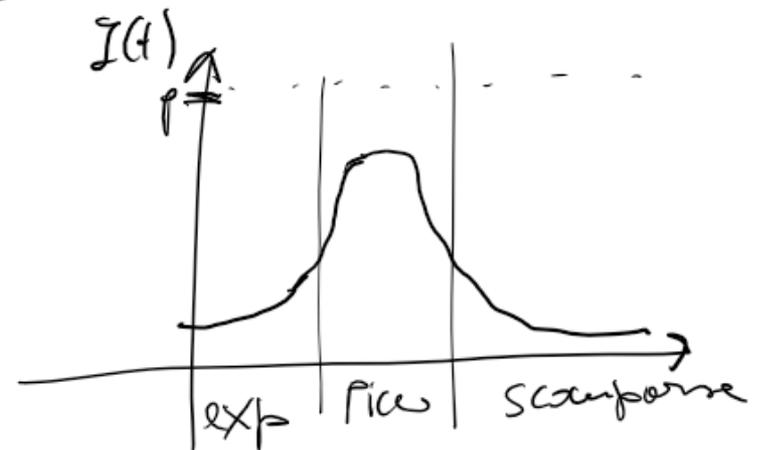
Esempio: contagi (iv)

Descrivere il possibile andamento delle infezioni $I(t)$

- Modello «S-I-R» (suscettibili – infetti - risolti)



... gli infetti, con un tasso μ , guariscono e diventano immuni alla malattia



Bussola

- Note introduttive
- Simulazione a eventi discreti: nozioni di base
- Simulazione a eventi discreti con AnyLogic
- Modellazione di sistemi dinamici
 - Relazioni causa-effetto e retroazioni
 - Modelli base di comportamento
- **Rappresentazione di sistemi dinamici**
 - Diagrammi causa-effetto
 - **Reti flussi-livelli**
- Simulazione di sistemi dinamici con AnyLogic
- Esempi notevoli di modellazione

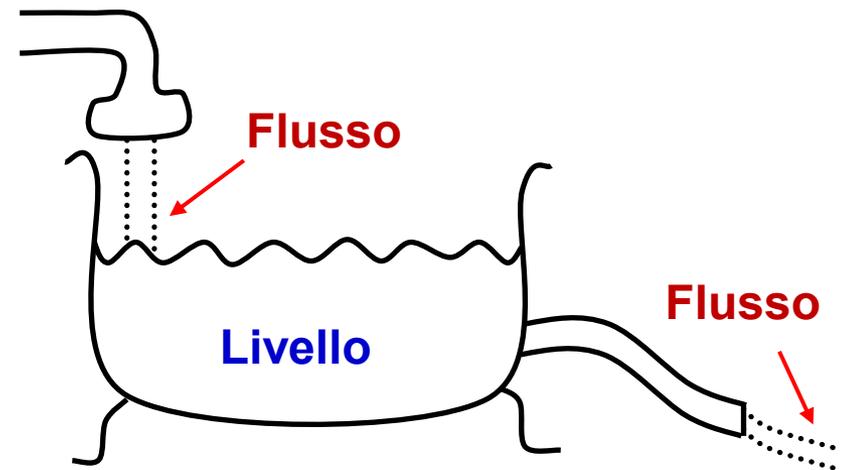
Limiti dei diagrammi causa-effetto

- Diagrammi causa-effetto (CLD): descrivono in modo efficace la **struttura** del sistema ma in modo **qualitativo**, prestandosi poco a cogliere gli aspetti quantitativi delle interazioni descritte
- CLD utili all'inizio, ma quando il modello comincia ad essere maturo, bisogna chiedersi:
 - il modello rappresenta bene il sistema allo studio, secondo i nostri scopi? [**validazione quantitativa** del modello, ad esempio per confronto con i dati del sistema reale]
 - quali sono gli effetti di variazioni nel modello (e quindi le variazioni attese nel sistema)? [**uso** del modello]
- **In definitiva: quali sono i livelli delle variabili nel tempo, come riprodotti dal modello? [= quale è la dinamica del sistema?]**
- **Bisogna passare ad un modello quantitativo!**



Variabili flusso e variabili livello

- Per una descrizione **quantitativa** e, quindi, per la **descrizione della dinamica** del sistema, alcune variabili dovrebbero indicare dei livelli, accumulando le variazioni (flussi) determinate da altre variabili
- È necessario distinguere:
 - **variabili di livello** (stock): rappresentano grandezze che evolvono nel tempo
 - **variabili di flusso** (flow): rappresentano variazioni nell'unità di tempo



Reti flusso-livello

Il formalismo delle reti flusso-livello (o Stock-Flow Network – **SFN**) permette di:

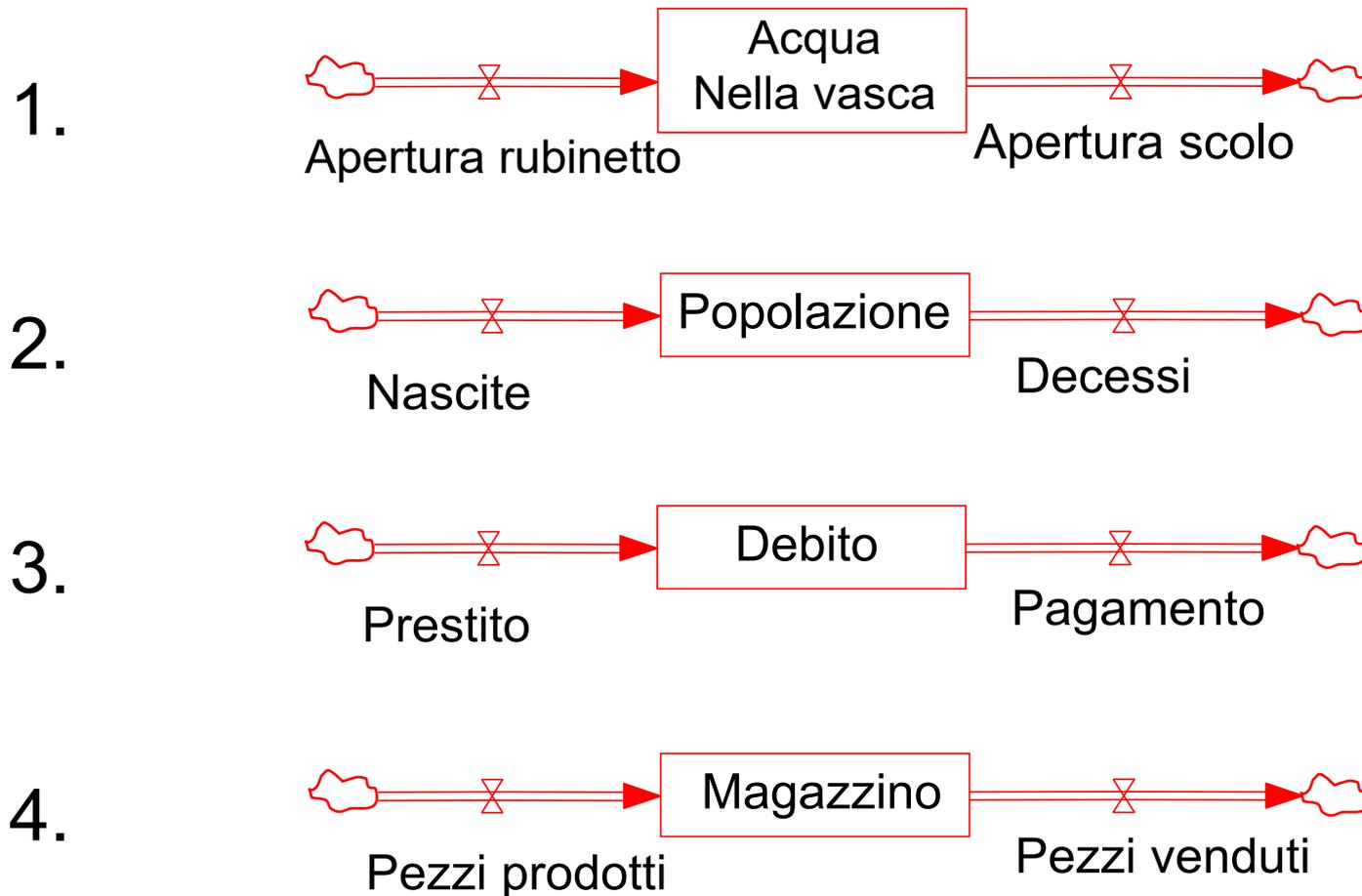
- identificare diversi tipi di variabili (livello, flusso, ausiliarie)
- descrivere **relazioni quantitative** tra le variabili del sistema
- implicitamente, descrivere le variabili, le loro interazioni, i relativi feedback (come i CLD)

Fornisce una **descrizione quantitativa** della **dinamica del sistema**

Formalismo SFN: base



Esempi SFN



La dinamica

Il formalismo SFN ha un significato quantitativo



- Ogni flusso è descritto come funzione del tempo
Flusso input = $I(t)$ Flusso output = $O(t)$
- Ad ogni livello è associata una funzione del tempo
Livello = $L(t)$
- Il formalismo SFN sintetizza la relazione

$$\frac{dL}{dt} = I(t) - O(t)$$

[**variazione** del livello **nel tempo**]

Equivalenze



$$\frac{dL}{dt} = I(t) - O(t)$$

$$L(t) = \int_{t_0}^t (I(s) - O(s)) ds$$

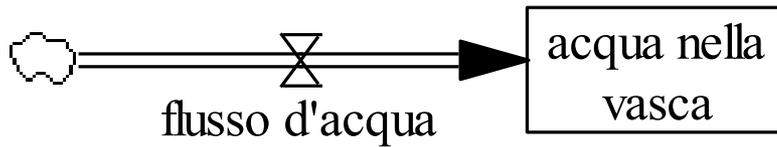
$$L(t + \Delta t) = L(t) + (I(t) - O(t))\Delta t \quad L(t) = \text{INTEG}(I(t) - O(t); L(t_0))$$

(discretizzazione: simulazione!)

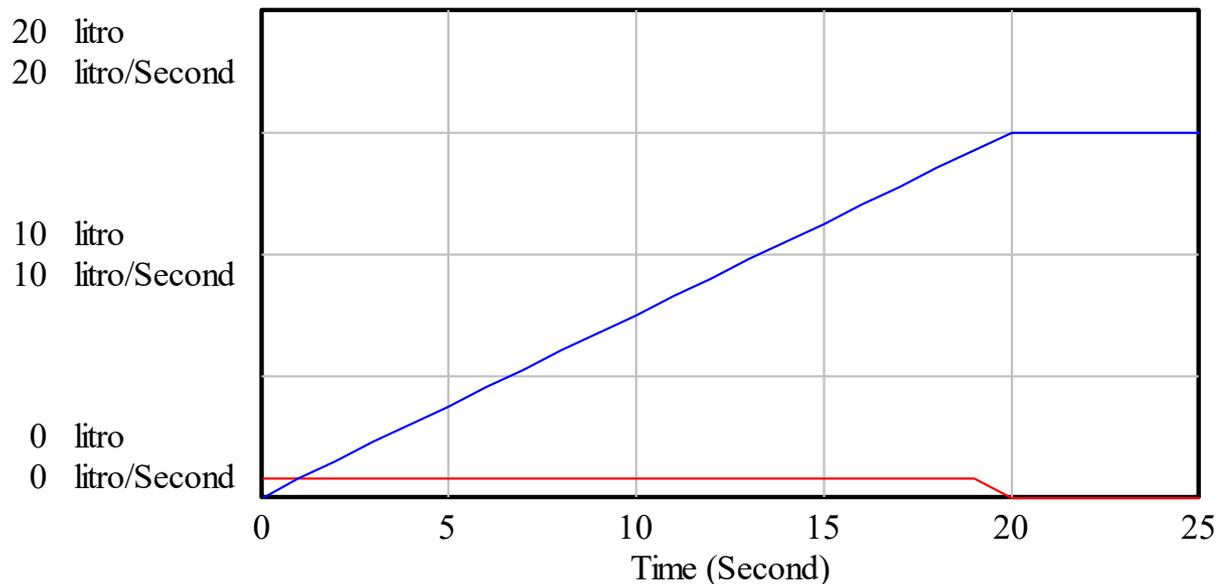
Sono modi **equivalenti** per esprimere la **dinamica del sistema** in modo quantitativo

Nota: le reti flussi-livelli sono adatte a variabili «continue»
(specificano valori per ogni t [o di Δt in Δt], non esistono t «speciali»)

La vasca da bagno



$$\equiv \text{Acqua nella vasca}(t + \Delta t) = \text{acqua nella vasca}(t) + \text{flusso d'acqua}(t) \cdot \Delta t$$



$$t \leq 20 : L(t) = \int_0^t 0.8 ds$$

$$= 0.8s \Big|_{s=0}^{s=t} - 0.8s \Big|_{s=0}$$

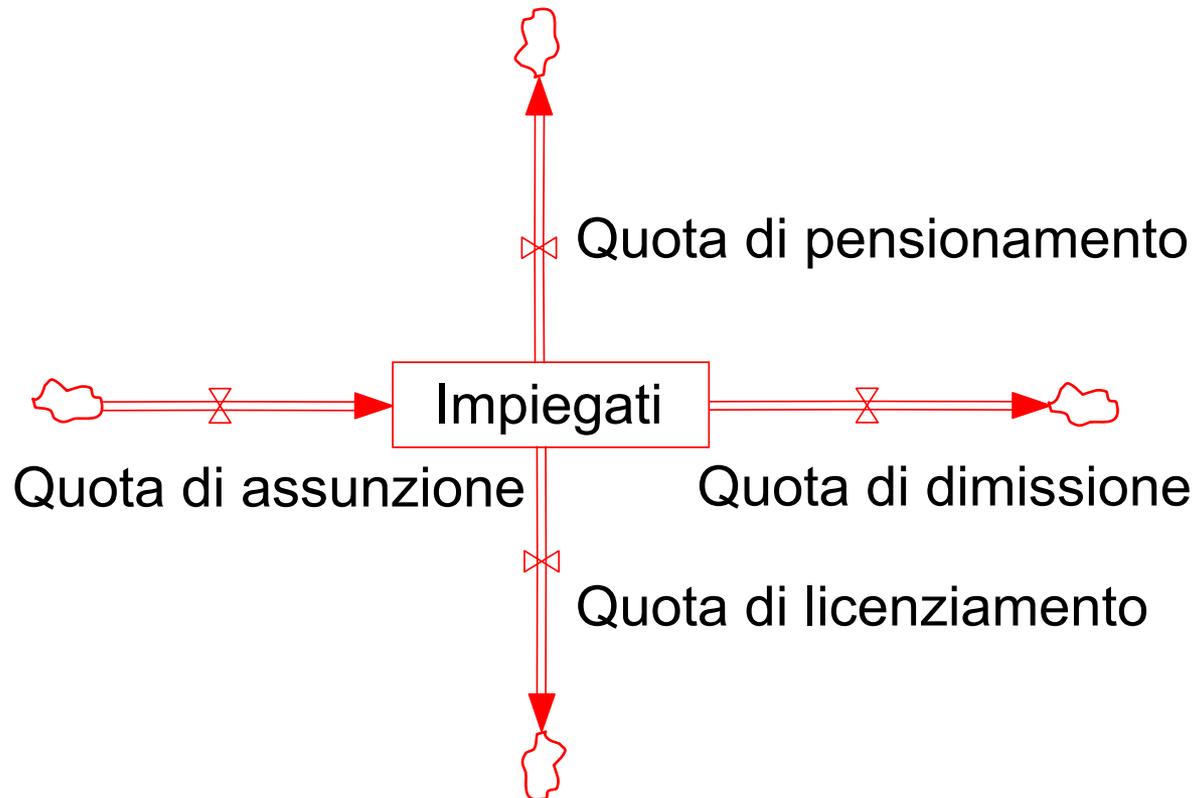
$$= 0.8t$$

$$t > 20 : L(t) = L(20) + \int_{20}^t 0 ds$$

$$= L(20)$$

acqua nella vasca : base ————— litro
 flusso d'acqua : base ————— litro/Second

Esempio: “Risorse umane”



$$\begin{aligned} \text{Impiegati}(\mathbf{t}+\Delta\mathbf{t}) &= \text{Impiegati}(\mathbf{t}) + [\text{Quota di assunzione}(\mathbf{t}) \\ &\quad - \text{Quota di dimissione}(\mathbf{t}) - \text{Quota di pensionamento}(\mathbf{t}) \\ &\quad - \text{Quota di licenziamento}(\mathbf{t})] \cdot \Delta\mathbf{t} \end{aligned}$$

Livello .vs. flusso

Importante distinguere livelli (fulcro delle equazioni) e flussi (elementi delle equazioni)

- **Livello**: variabile di “accumulazione”
 - Sono **quantità** che vengono accumulate/erose
 - Rappresentano lo **stato** del sistema
 - Forniscono la **memoria** del sistema

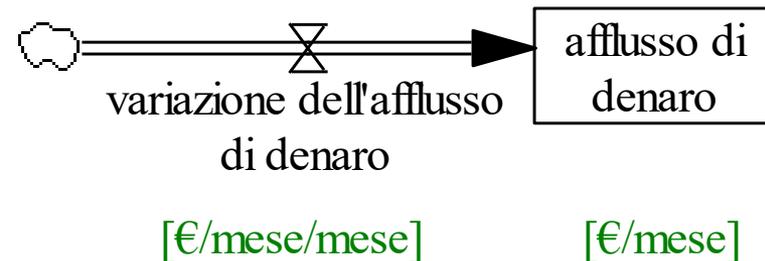
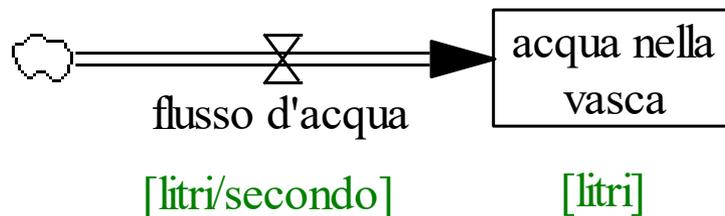
Esempi: popolazione, riserve di acqua o petrolio, debito, CO₂ nell’atmosfera; ma anche rabbia, frustrazione, motivazione, autostima etc.

- **Flusso**: variabile di “variazione”
 - Sono le **velocità** di accumulazione/erosione
 - Corrispondono ad **azioni** compiute nel sistema

Esempi: nascite, morti, flusso di cassa, emissione di gas etc.

Livello .vs. flusso: in pratica

- Schema **linguistico** [sistema come descrizione]
 - Livello: nomi
 - Flusso: verbi
- Immaginiamo di **fermare** il sistema
 - ciò che osserviamo: livello (l'accumulazione rimane)
 - ciò che scompare: flussi (l'azione si ferma)
- Le variabili hanno **unità di misura** (descrizione quantitativa!)
 - Livello: [U]
 - Flusso: [U/t]



Regole di formalizzazione SFN

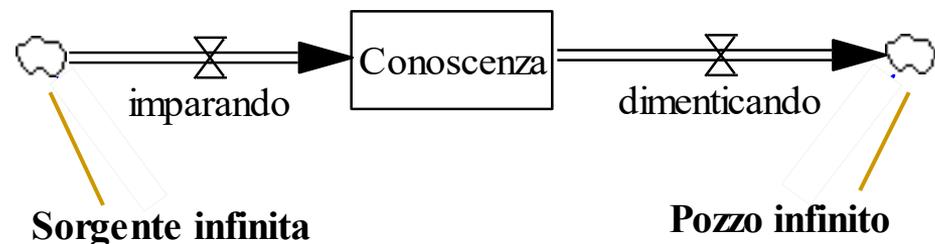
■ Consistenza dimensionale

- i flussi entranti e uscenti da uno stock devono avere la stessa unità di misura: quella dello stock divisa per l'unità di tempo



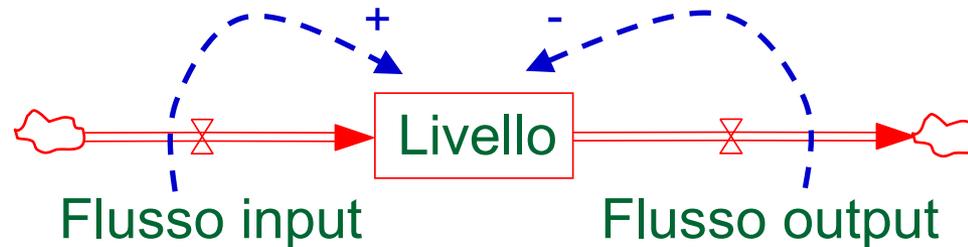
■ Legge di **conservazione del flusso**

- tutti i flussi giungono da qualche sorgente (riducendo il livello di qualche stock) e vanno da qualche parte del modello (incrementando il livello di qualche stock)
- i flussi di cui non ci interessa indicare la sorgente o il pozzo (dipende dalla definizione del dominio di interesse) provengono o si riversano in sorgenti o pozzi di capacità infinita



Formalismo SFN e relazioni causali

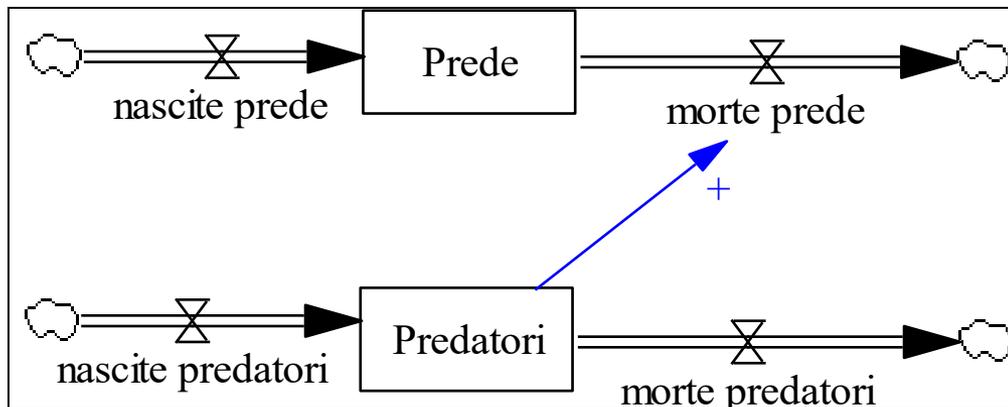
- La presenza di flussi indica, in modo implicito, delle **relazioni di causa-effetto**



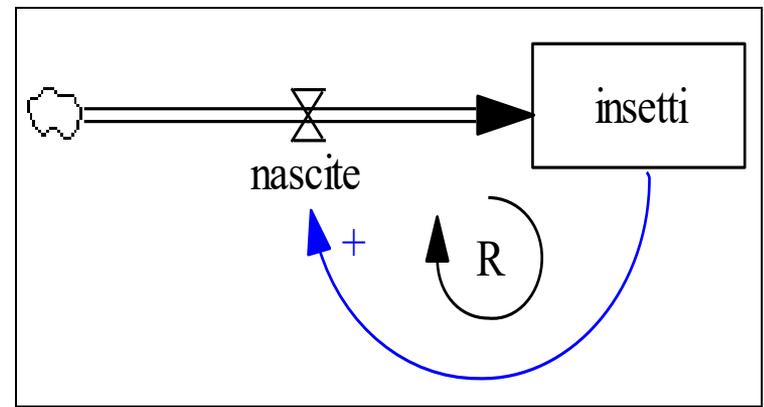
- Formalismo SFN si può anche prestare all'analisi qualitativa (individuare retroazioni, dinamiche base, possibili interazioni etc.)

Formalismo SFN: connettori

- Il **connettore** trasmette informazioni concernenti lo stato delle variabili del sistema (introducono **ulteriori relazioni causa-effetto**).
- Queste informazioni saranno utilizzate **per determinare i flussi**. Pertanto i connettori non possono entrare in uno stock: le uniche relazioni causali verso uno stock sono quelle implicite dei flussi (*se determina variazioni di un livello è un flusso, per definizione*)
- Formalismo delle relazioni causali dei diagrammi causa-effetto (freccie polarizzate con eventuali feedback)

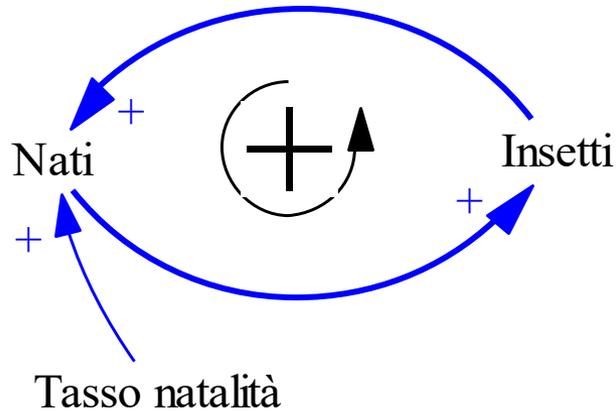


Ad es.: $morte\ prede(t) = \beta\ Predatori(t)$

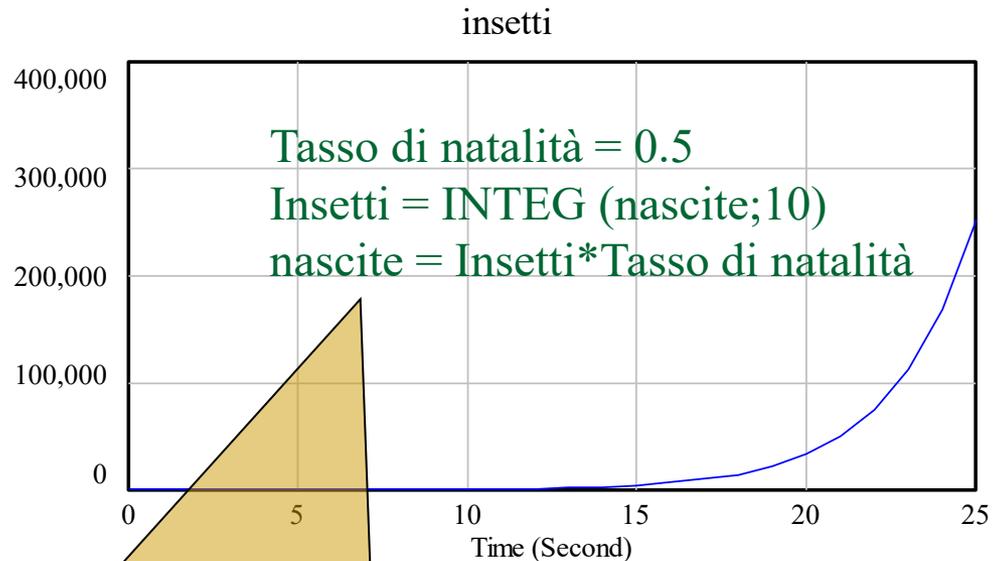
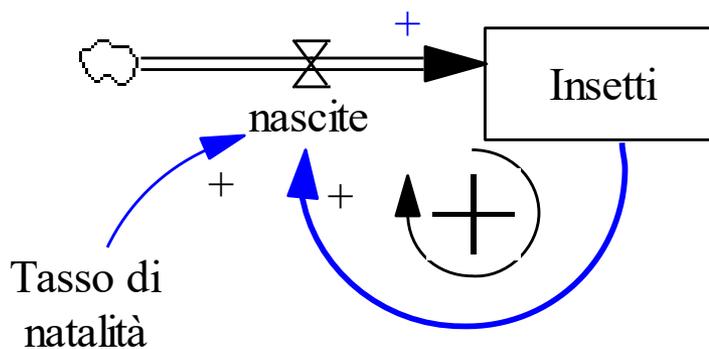


Ad es.: $nascite(t) = \alpha\ insetti(t)$

Traduzione di modelli CLD (i)

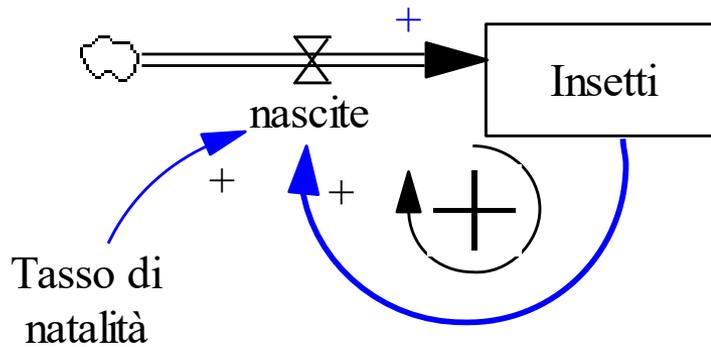


- Identificare degli **stock**
 - numero di insetti [unità]
- Identificare i **flussi**
 - Nati [unità/mese]
- Identificare **costanti** (non subiscono effetti):
 - tasso di natalità [%]



Corrisponde a un'equazione differenziale....

L'equazione differenziale...



tempo $\leadsto t$
Insetti $\leadsto I(t)$
tasso di natalità $\leadsto r$
nascite $(t) \leadsto m(t) = rI(t)$

$$\begin{cases} \frac{dI(t)}{dt} = m(t) = rI(t) \\ I(0) = 10 \end{cases}$$

$$\Rightarrow \begin{cases} I(t) = C e^{rt} \\ C = 10 \end{cases}$$

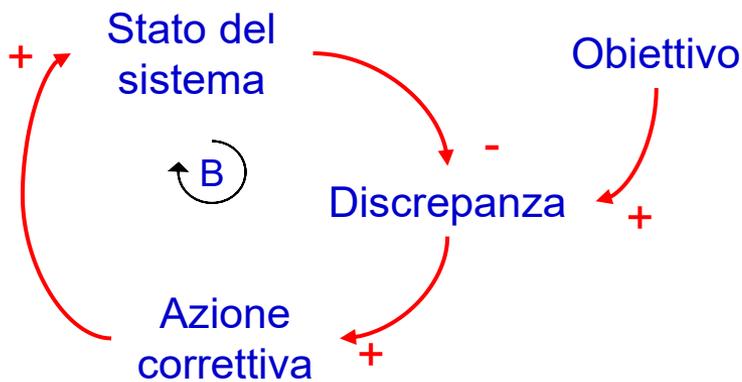
DINAMICA $I(t) = 10 e^{rt}$

Intro a CLD e SFN con AnyLogic

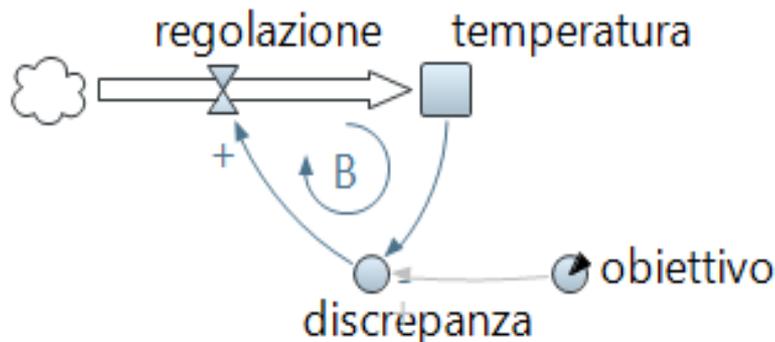
- Esercizio 0, Parte I (dinamica semplificata di una popolazione)
 - Attenzione: una relazione di causa-effetto deve corrispondere alla partecipazione della variabile cuasa nella formula che determina la variabile (e viceversa)

Traduzione di modelli CLD (ii)

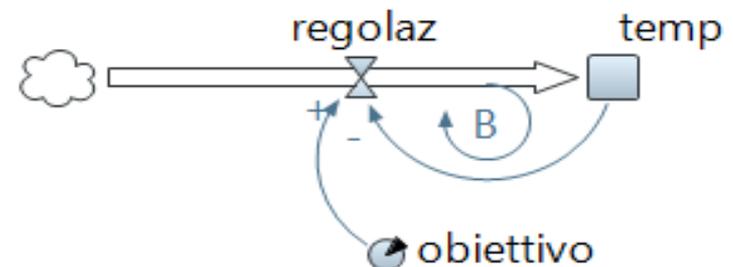
Goal seeking



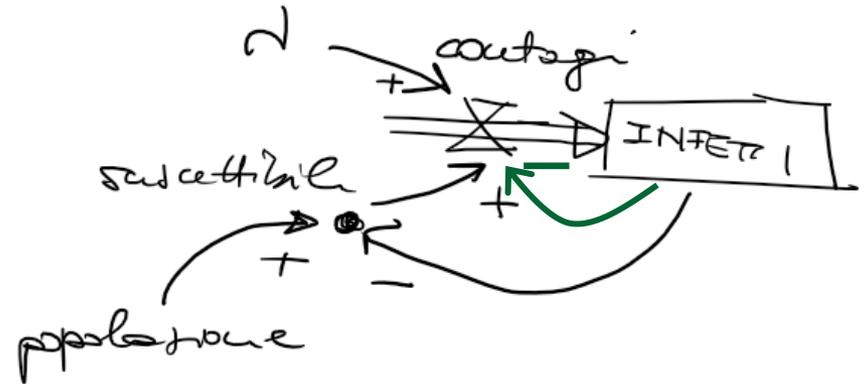
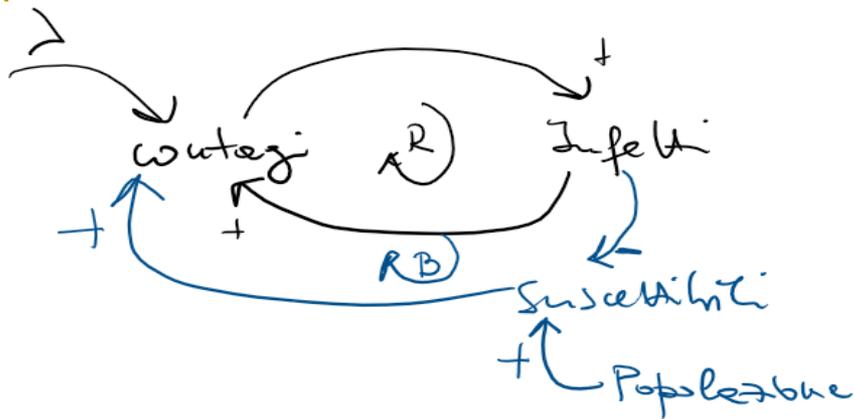
- *Stato del sistema*: stock
- *Azione correttiva*: flusso
- *Obiettivo*: costante
- *Discrepanza?*
 - Non può essere stock o flusso!
 - È una **variabile ausiliaria** che **esplicita** la differenza tra obiettivo e stato
 - Può essere omessa (“nascosta” nell’espressione del flusso): non influenza la dinamica!



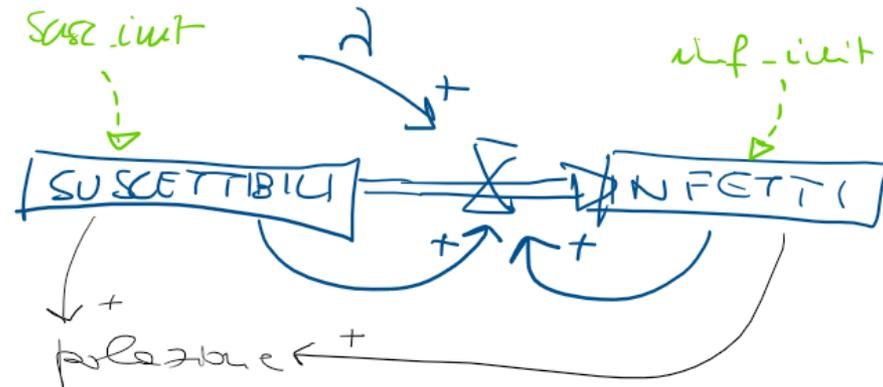
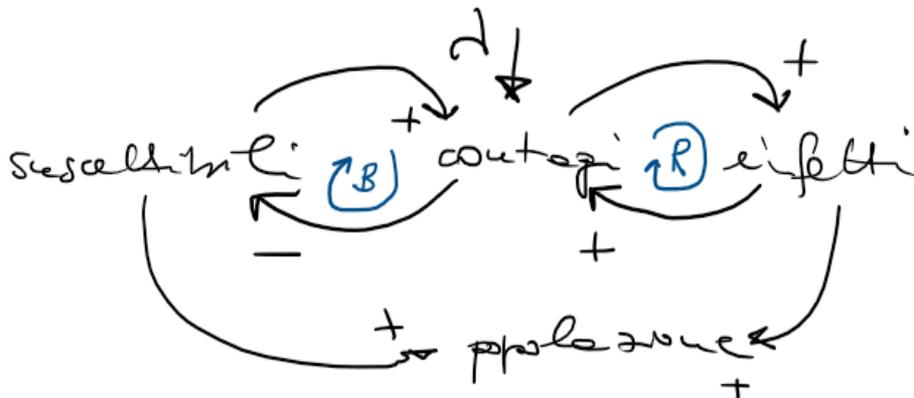
≡



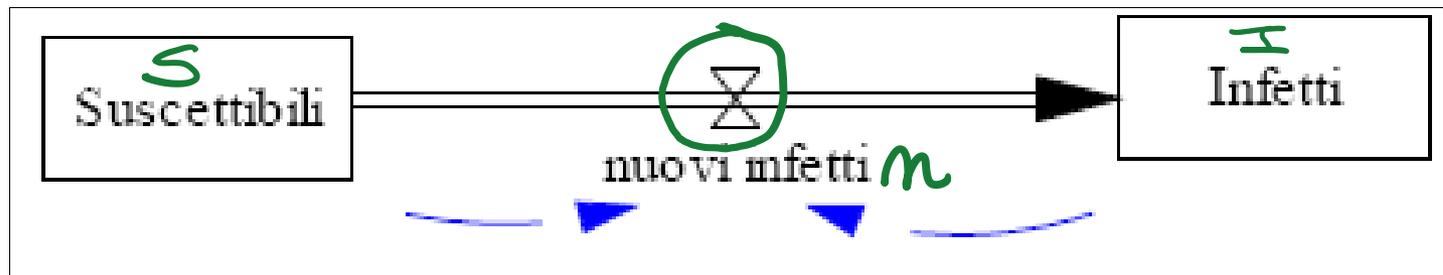
Traduzione di modelli CLD (iii)



- **Attenzione!** non possiamo rappresentare «suscettibili» come stock! «i connettori (da *INFETTI* e da *popolazione*) non entrano in uno stock»
- \Rightarrow *suscettibili* è una **variabile ausiliaria** («calcolata»)
- In alternativa, una rappresentazione con un **modello equivalente**



Sistemi di equazioni differenziali



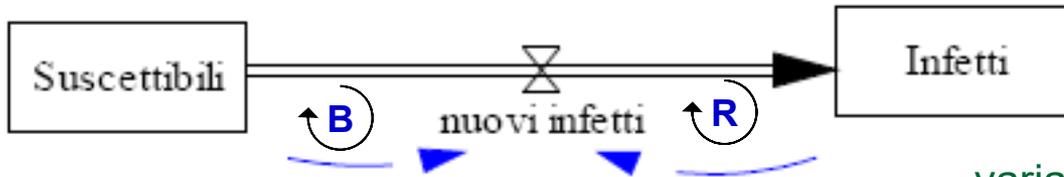
$$\begin{cases} S'(t) = -m(t) \\ I'(t) = +m(t) \\ m(t) = f(S(t), I(t)) \\ S(0) = S_0 \\ I(0) = I_0 \end{cases}$$

soluzione...

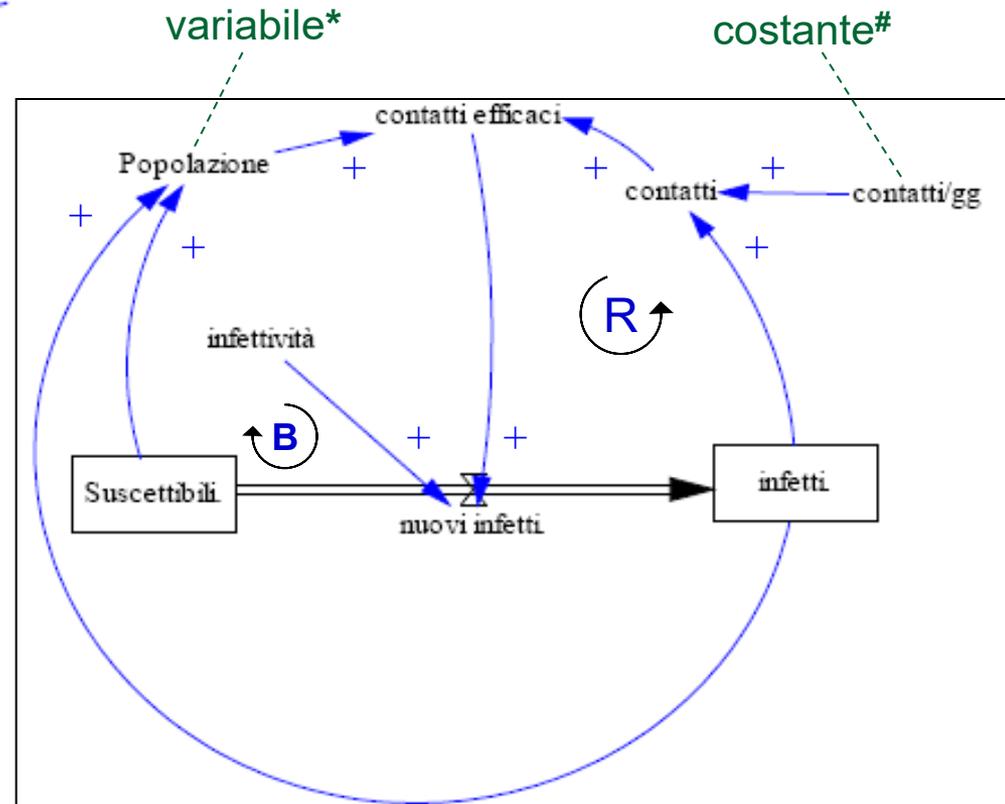
$S(t) = \dots$
 $I(t) = \dots$
per via analitica
se "possibile"

simulazione
numerica

Variabili ausiliarie e costanti



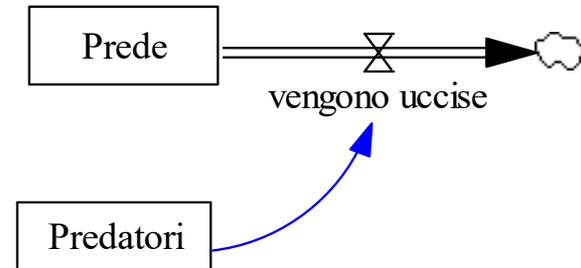
- Permettono di **esplicitare delle funzioni** altrimenti complesse e poco chiare: migliorano la leggibilità
- Utilizzano il **formalismo dei diagrammi causa-effetto**
- Arricchiscono il formalismo «estetico» senza aggiungere capacità descrittiva della dinamica: la **dinamica è invariata**



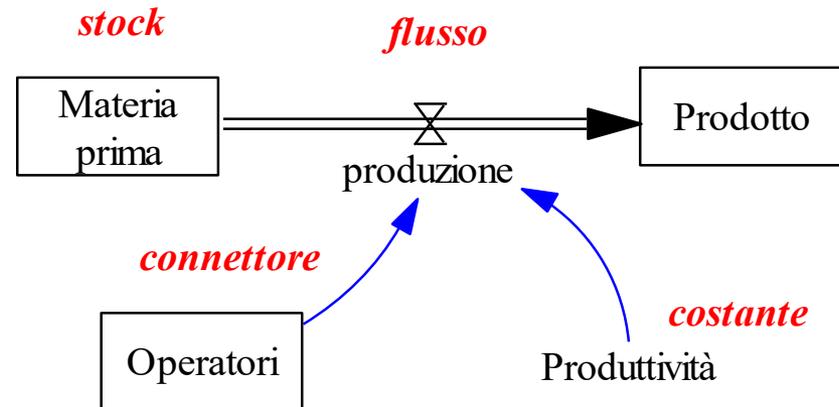
- *: la dinamica è influenzata da altri elementi
- #: la dinamica **non** è influenzata da altri elementi

Traduzione di frasi

- Le prede vengono uccise dai predatori



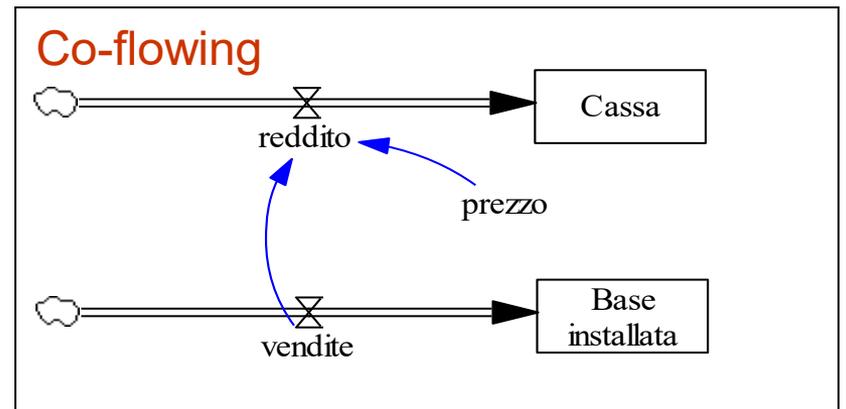
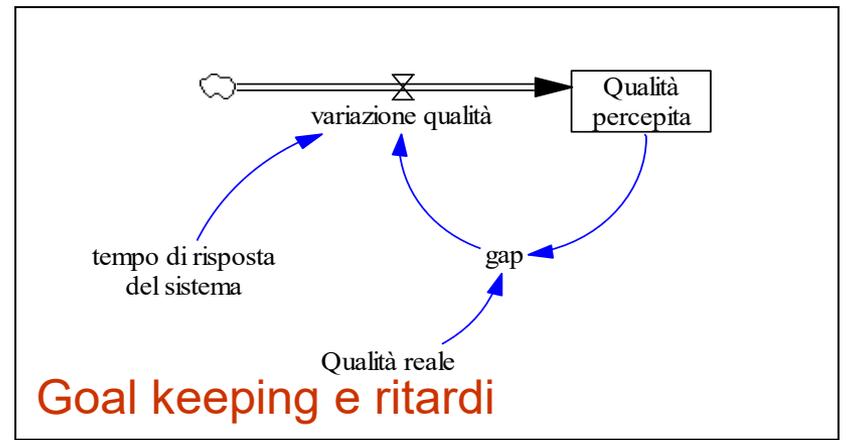
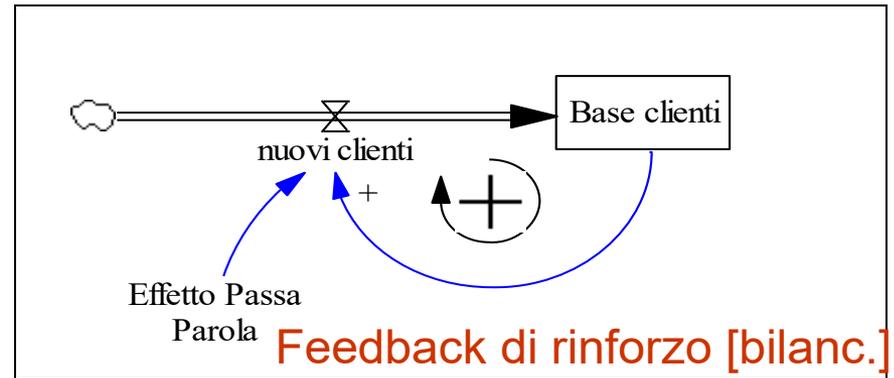
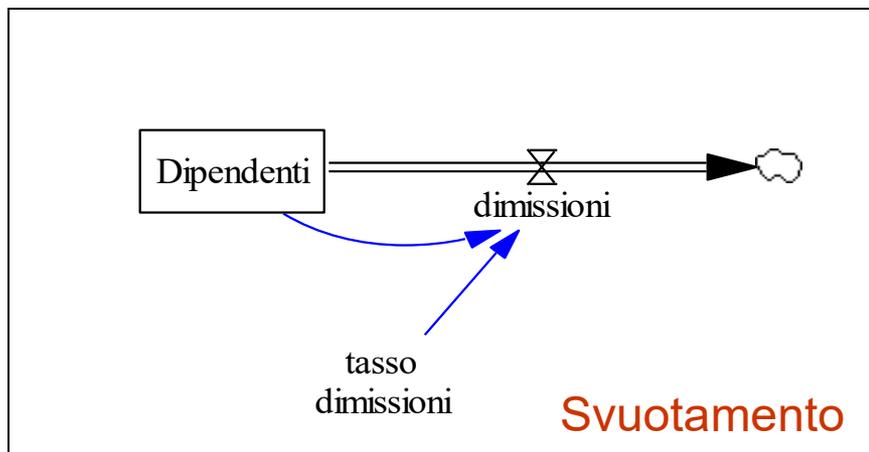
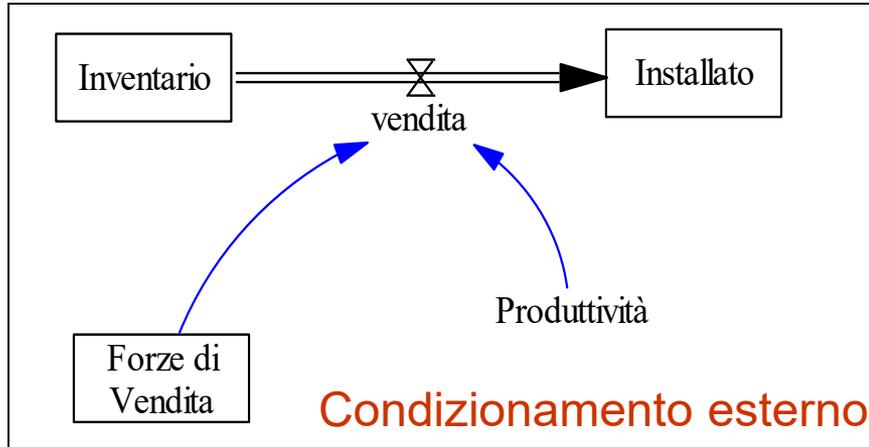
- Il flusso della produzione dipende dalla disponibilità di materia prima, dal numero di operatori e dalla loro produttività



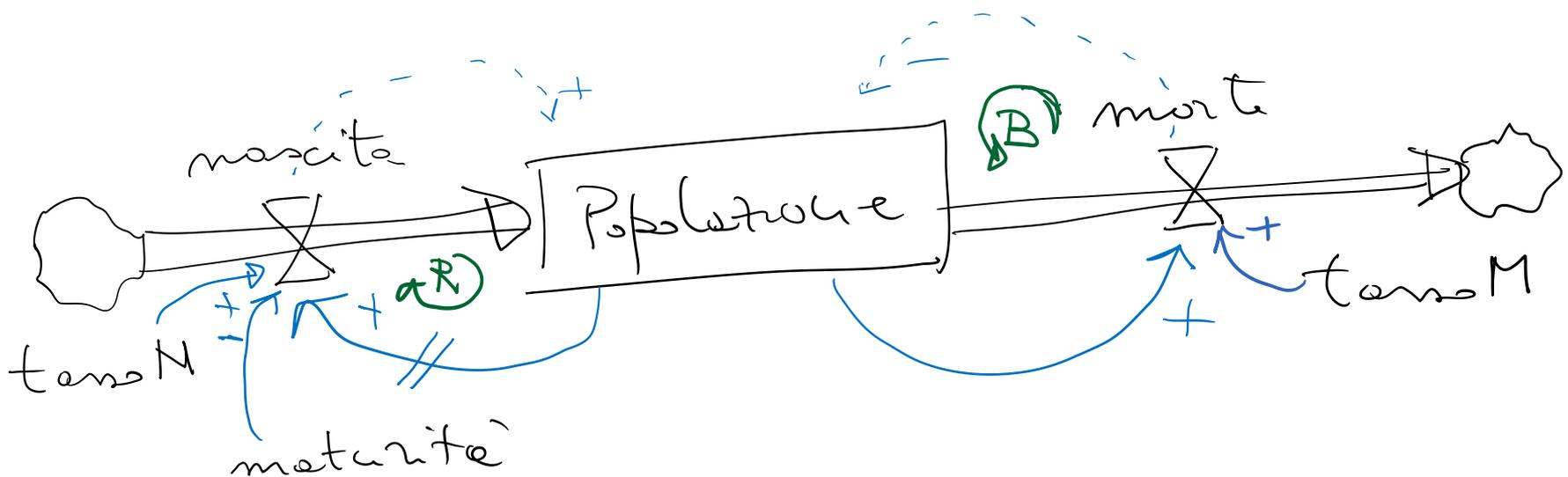
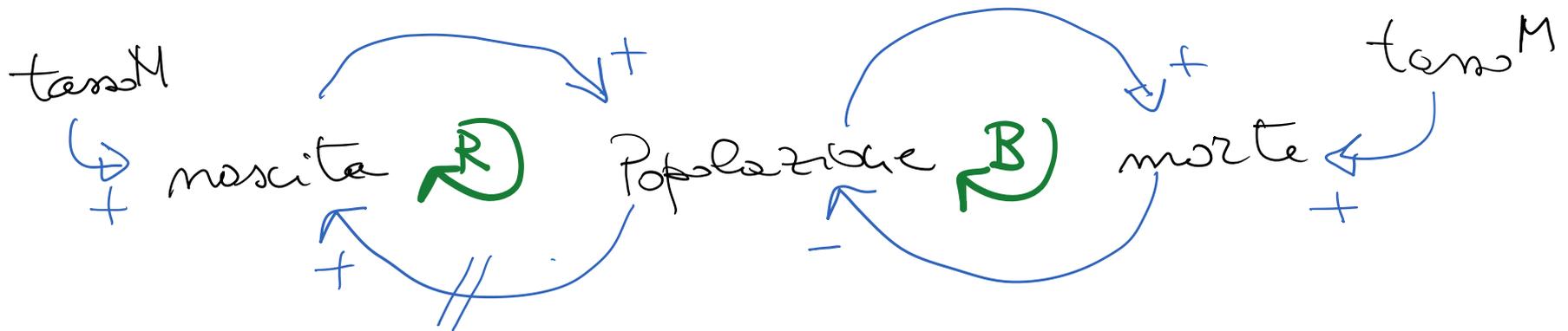
- Attenzione:** si tratta di modelli parziali (non è rispettata, ad esempio, la conservazione dei flussi)

Esempi notevoli

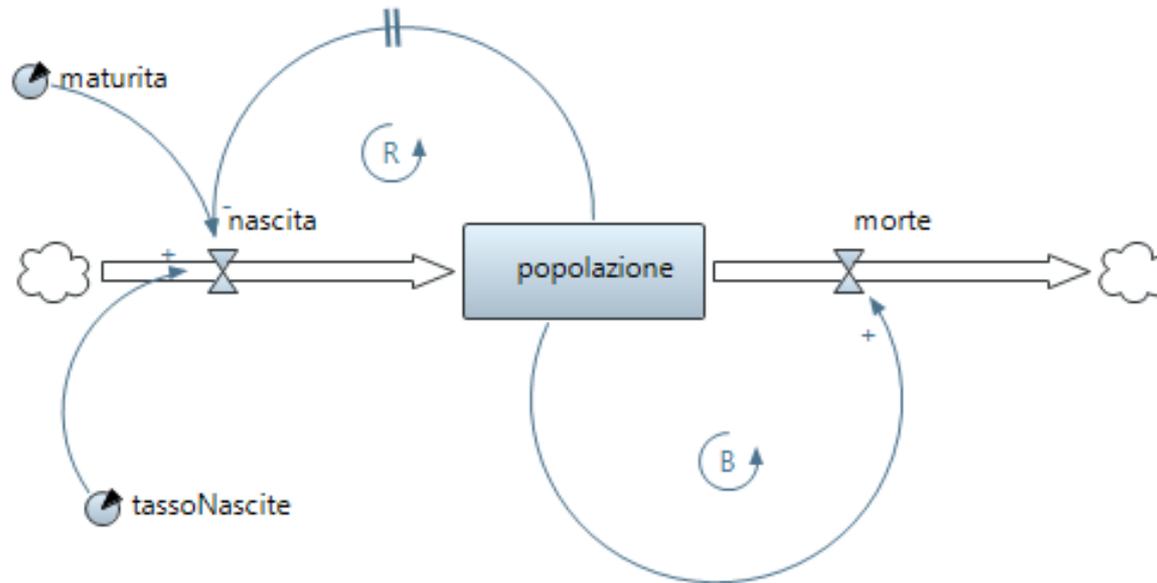
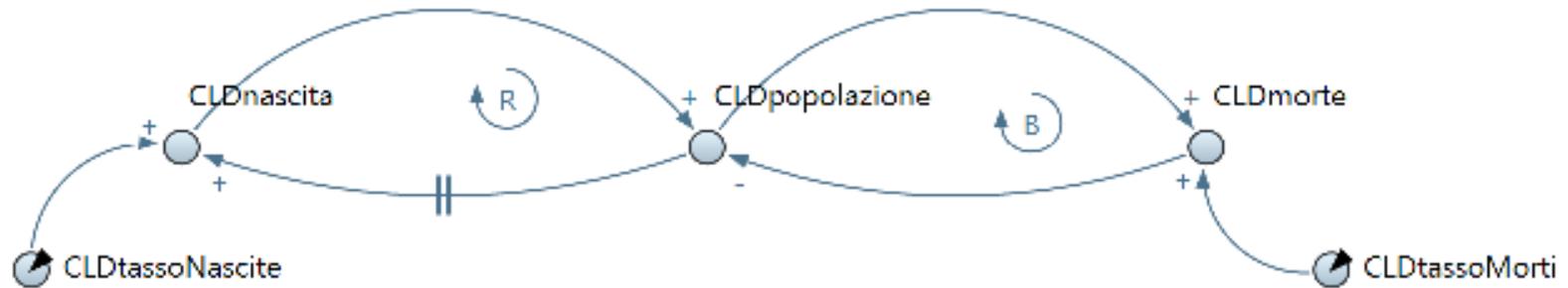
Nota: le polarità facilitano la leggibilità ma non entrano nelle equazioni della dinamica (consigliate, non obbligatorie)



Esempio: dinamica popolazione



Esempio: dinamica pop. (AnyLogic)



Esempio: sistema delle vendite

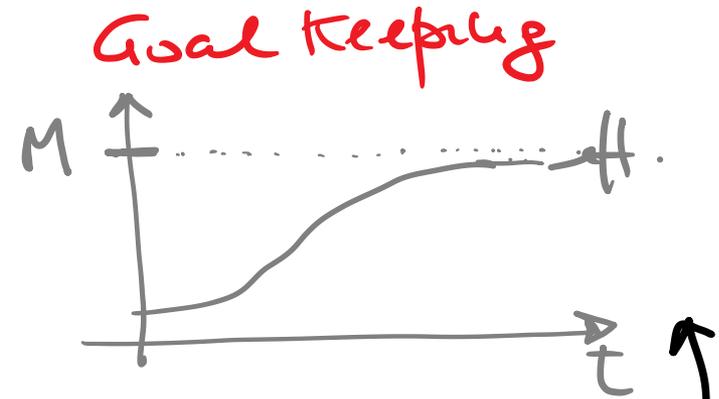
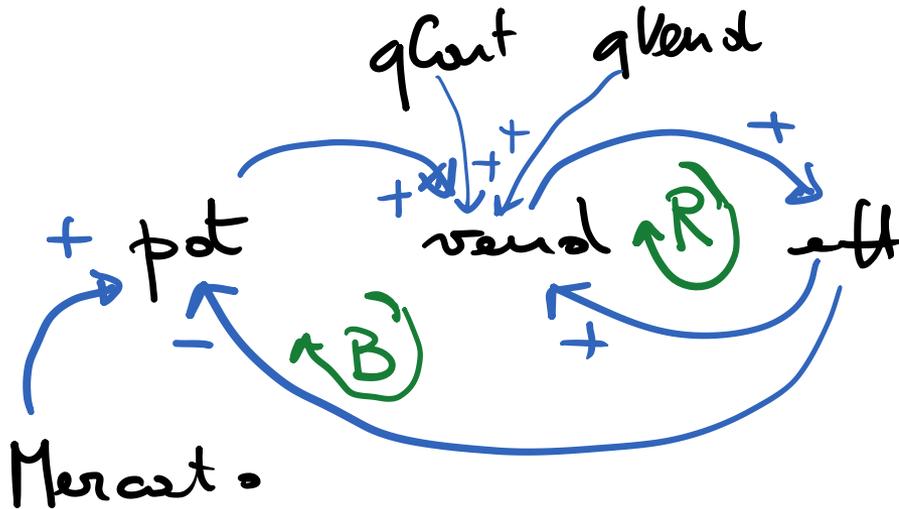
1. Rappresentare un sistema di vendite alimentate dal passaparola tra clienti effettivi e clienti potenziali
2. e determinare le dinamiche del numero di clienti effettivi e clienti potenziali

a partire da informazioni sul mercato totale, il numero di clienti iniziali, la possibilità di raggiungere le persone potenzialmente interessate (quota di contatto), l'efficacia dei contatti tra clienti potenziali ed effettivi ai fini dell'acquisizione di nuovi clienti (quota di vendite)

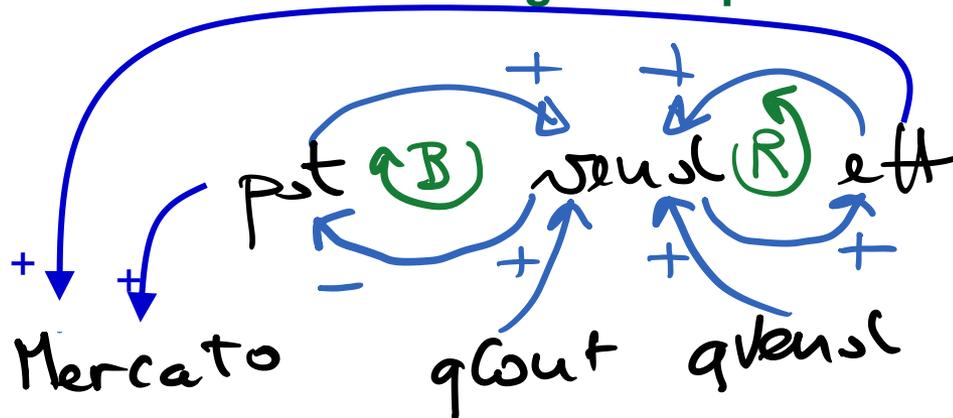
Nota: utilizzeremo un modello di diffusione del tipo definito da F.M. Bass (Bass Diffusion Model). Vedi anche tutorial di AnyLogic.

Esempio: sistema delle vendite

- (1) Diagramma causa-effetto I



vogliamo «pot» come stock

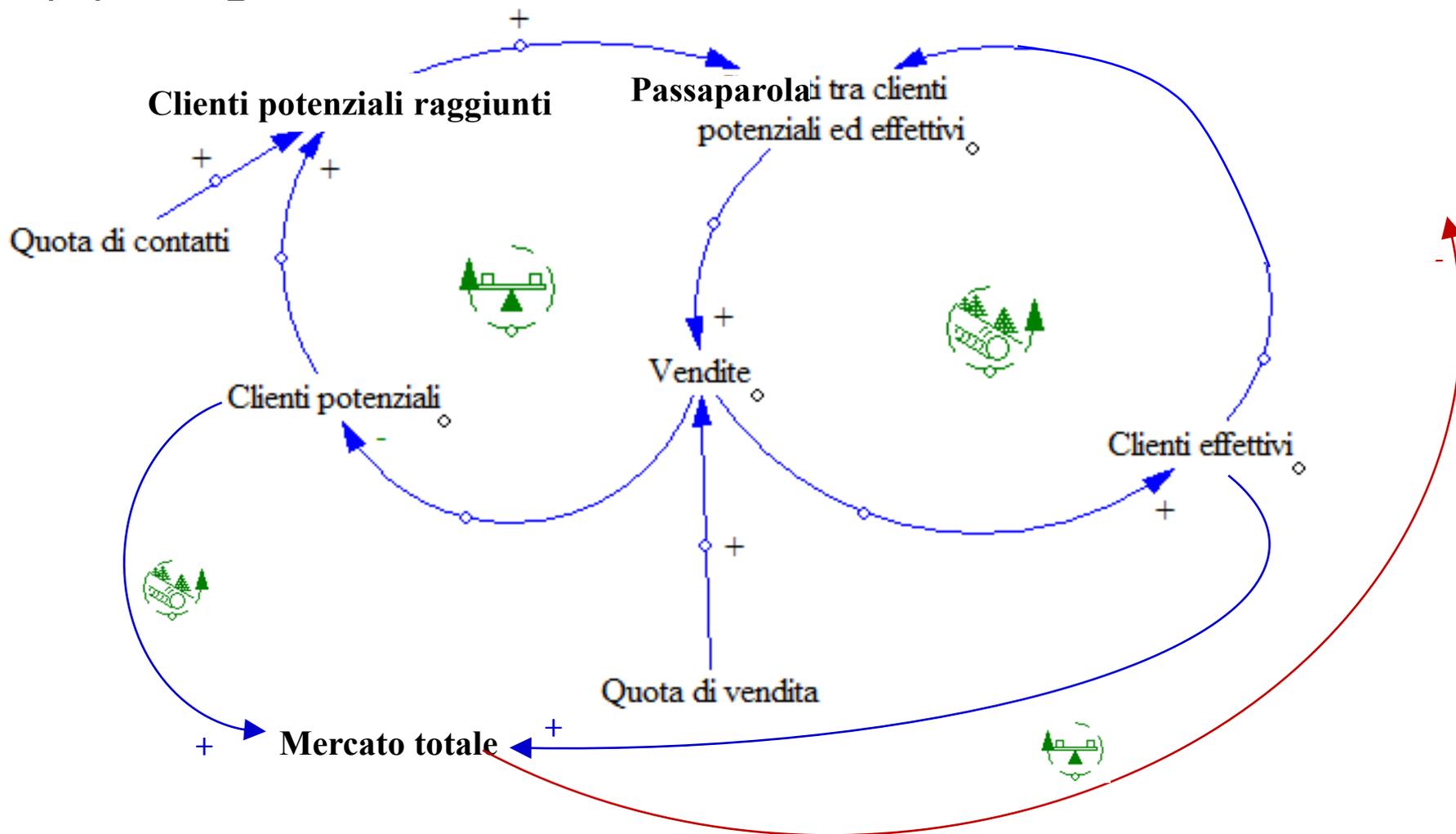


(stessa dinamica)

(se Mercato è quello iniziale e rimane costante pari a $pot + eff$)

Esempio: sistema delle vendite

■ (1) Diagramma causa-effetto II



Esempio: sistema delle vendite

I clienti effettivi e potenziali sono riferiti a un mercato complessivo. Inoltre si evidenzia l'importanza non solo del numero assoluto di clienti effettivi, ma della loro prevalenza (percentuale) rispetto al mercato, il che attiva due ulteriori feedback di rinforzo e bilanciamento.

Si evidenziano inoltre i contatti per rendere visibile l'occasione del passaparola.

Nello schema dato la quota di contatto agisce sui contatti tra potenziali clienti: si immagina infatti che la quota di contatti esprima una misura della "socialità" del sistema ("temperatura" del sistema, *distanza sociale*), più che della capacità dei clienti effettivi di creare occasioni per parlare con i clienti potenziali.

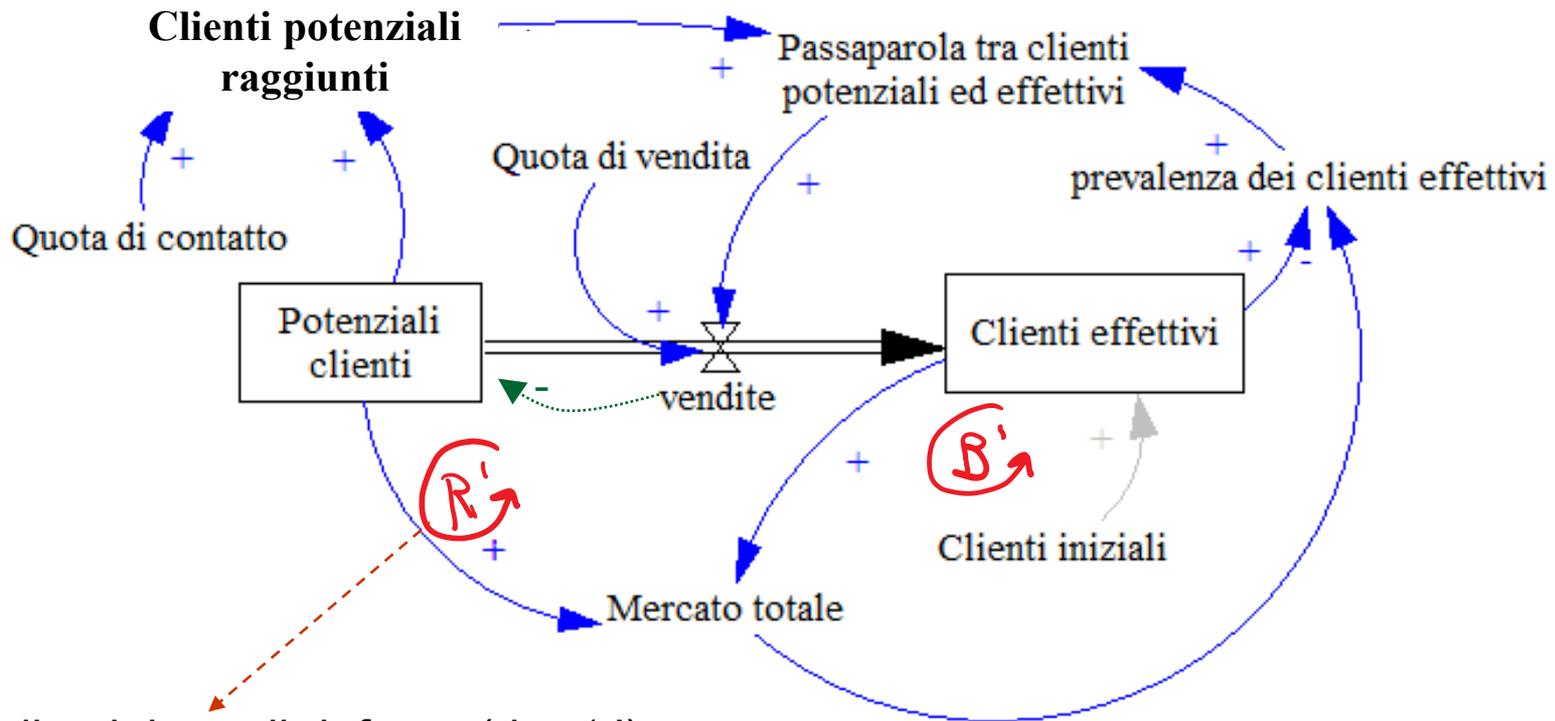
In alternativa, per un sistema diverso, la quota di contatti potrebbe avere effetto direttamente sui contatti tra clienti potenziali e effettivi, col significato misura delle persone con le quali i clienti effettivi scambiano opinioni.

Sono modelli alternativi, che mettono l'accento su cose diverse, a seconda di cosa sia importante per il modellatore: rappresentare la "temperatura" del mercato (dovuta ad esempio a fattori sociologici), oppure la pervasività dei clienti effettivi (entusiasti del prodotto).

Notare che, da un punto di vista quantitativo, a parità di «*Quota di contatti*» e se, ad esempio, utilizziamo delle moltiplicazioni come funzioni per determinare le vendite, non cambia nulla (vedi implementazione in AnyLogic).

Esempio: sistema delle vendite

■ (2) Rete livelli-flussi



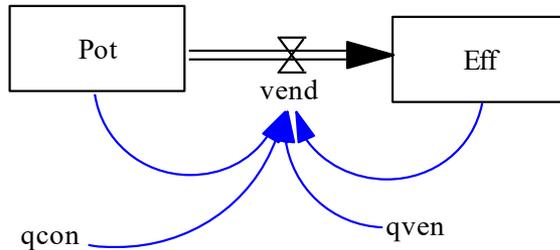
Feedback loop di rinforzo (due '-')

Potenziali clienti →+ Mercato totale →- prevalenza dei clienti effettivi →+

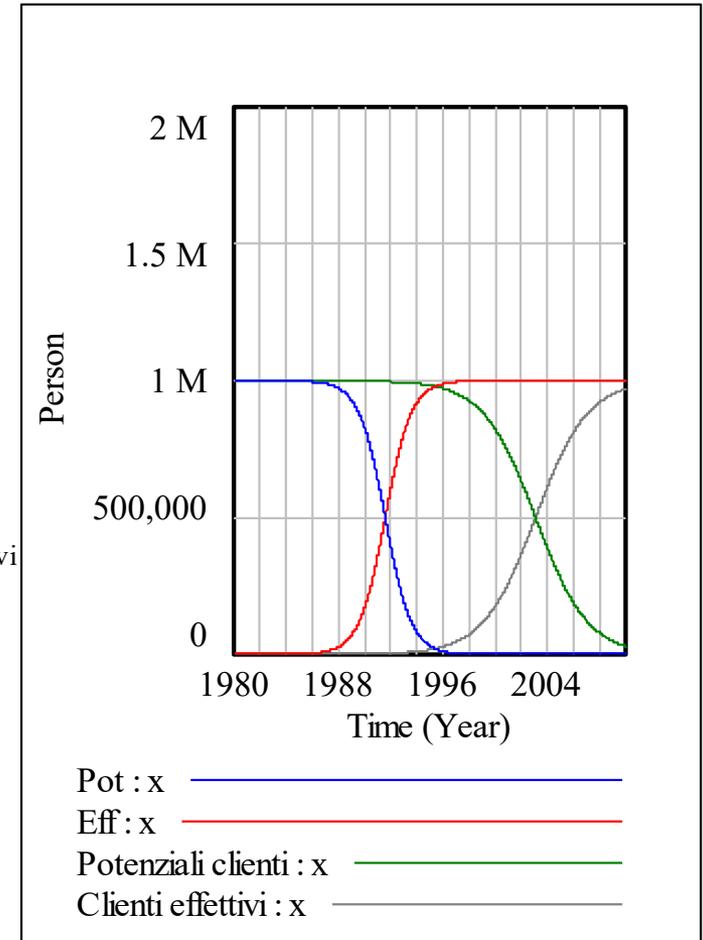
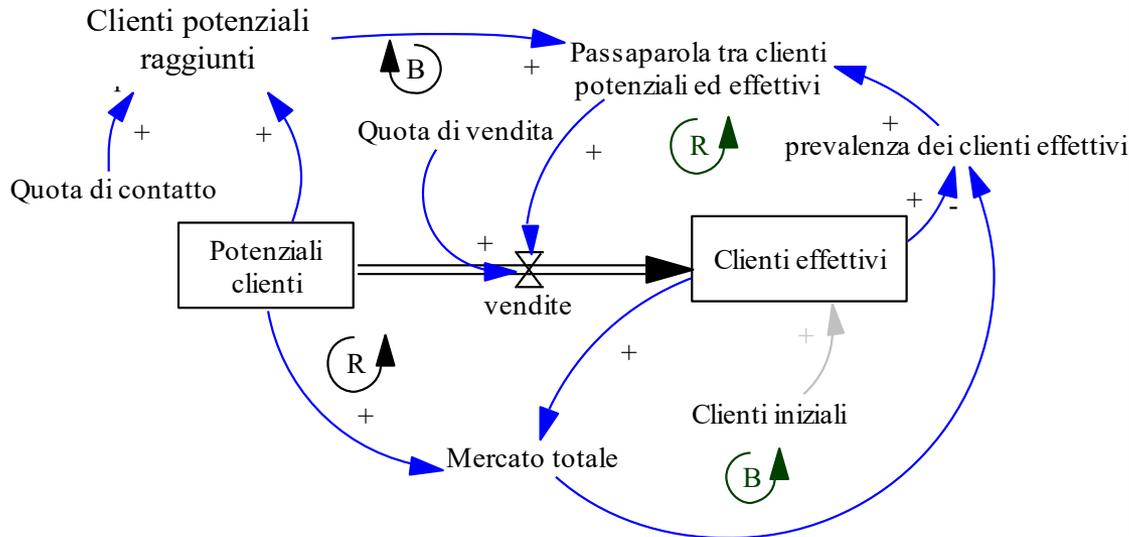
Passaparola tra clienti potenziali ed effettivi →+ **vendite** →- **Potenziali clienti**

Esempio: sistema delle vendite

Modello Base



Modello esplicativo



Nota: con parametri e formule coerenti le dinamiche coincidono!

VenditeSFN.confronto.mdl

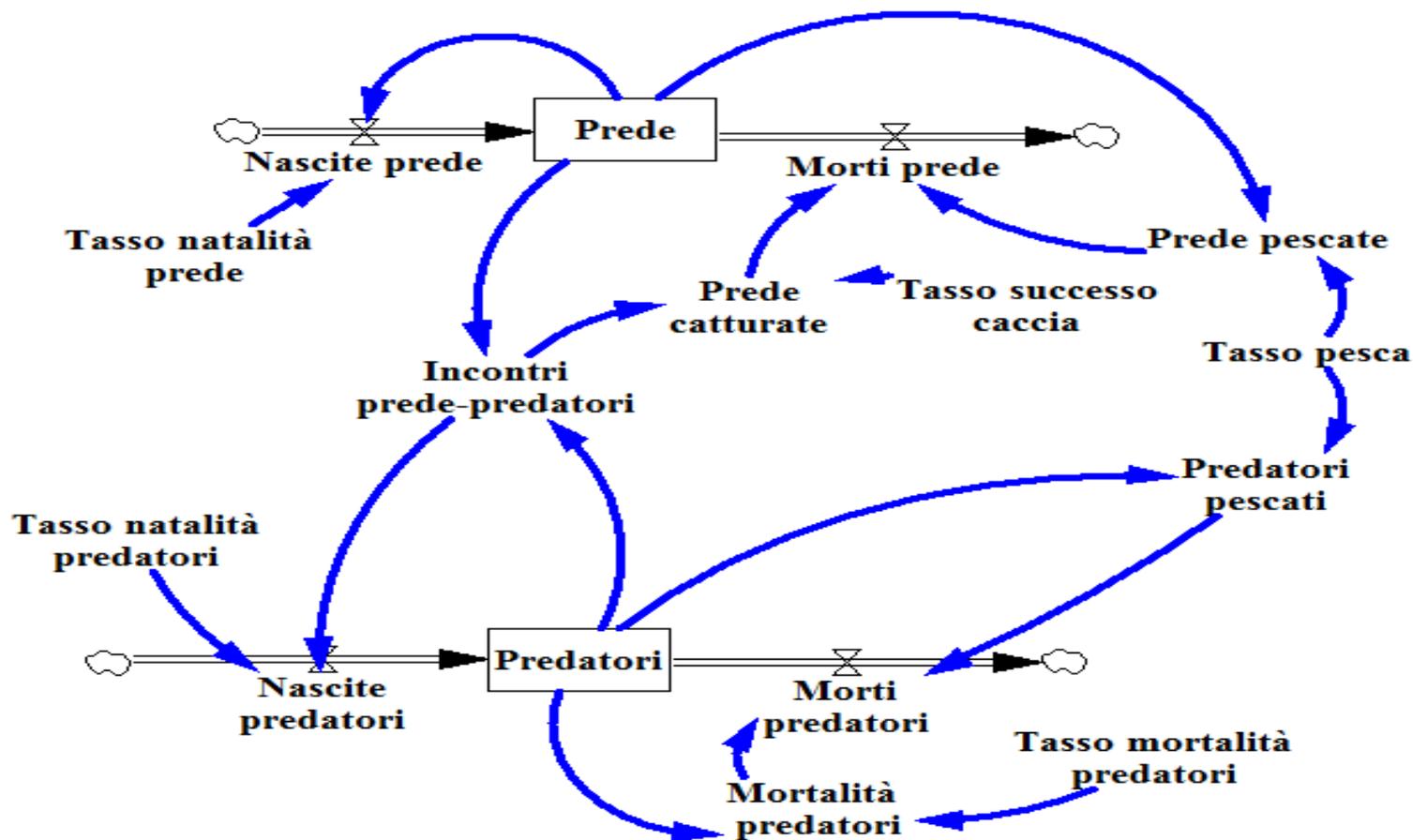
Esempio: preda-predatore

Si vuole studiare l'andamento nel tempo delle popolazioni di prede e predatori marini in un sistema nel quale:

- le prede e i predatori si riproducono con tassi annuali noti
- le prede sono cacciate dai predatori e la morte naturale delle prede è trascurabile
- i predatori sono soggetti a morte naturale, secondo un tasso annuale noto
- sia le prede che i predatori sono pescati con un tasso noto

Il modello preda-predatore è stato sviluppato da Vito Volterra (matematico, 1860-1940) sulla base delle osservazioni dello zoologo Umberto D'Ancona relative alla pesca nel nord dell'Adriatico in diversi anni.

Esempio: preda-predatore - SFN



[preda-predatore.mdl](#)

Esempio: preda-predatore (prede)

- Le prede nascono
 - $Nascite\ Prede(t) = Tasso\ natalità\ prede \cdot Prede(t)$
- Le prede possono morire o perché vengono pescate o perché vengono catturate da predatori
 - $Morti\ Prede(t) = Prede\ Catturate(t) + Prede\ Pescate(t)$
- Le prede vengono catturate
 - $Prede\ Catturate(t) = Tasso\ successo\ caccia \cdot Incontri\ Prede\ Predatori(t)$
dove
 - $Incontri\ Prede\ Predatori(t) = Prede(t) \cdot Predatori(t)$
- Le prede vengono pescate
 - $Prede\ Pescate(t) = Tasso\ pesca \cdot Prede(t)$
- I predatori nascono (il tasso di nascita dipende dalla disponibilità di prede e quindi dal numero di incontri prede-predatori)
 - $Nascite\ Predatori(t) = Tasso\ natalità\ predatori \cdot Incontri\ Prede\ Predatori(t)$

Nota: si ritiene trascurabile la morte naturale delle prede

Esempio: preda-predatore (predatori)

- I predatori nascono
 - $Nascite\ Predatori(t) = Tasso\ natalità\ predatori \cdot Incontri\ Prede\ Predatori(t)$
- I predatori possono morire o perché vengono pescati o di morte naturale
 - $Morti\ Predatori = Mortalità\ Predatori(t) + Predatori\ Pescati(t)$
- I predatori vengono pescati
 - $Predatori\ Pescati(t) = Tasso\ pesca \cdot Predatori(t)$
- I predatori muoiono naturalmente
 - $Mortalità\ Predatori(t) = Tasso\ Mortalità\ predatori \cdot Predatori(t)$

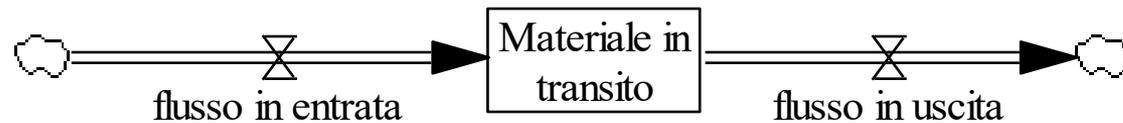
Ritardi

- I ritardi sono inevitabili ed estremamente importanti per la determinazione delle dinamiche dei sistemi
- I ritardi hanno effetti **negativi**:
 - determinano instabilità e oscillazioni
 - rendono più complesse le dinamiche
 - esempio: la solita doccia ...
- I ritardi hanno effetti **positivi**:
 - permettono una visione più chiara delle dinamiche in input
 - permettono di filtrare i rumori di fondo
 - esempio: se ritardassimo la decisione di variare la temperatura dell'acqua nella solita doccia ...

Tipi di ritardo

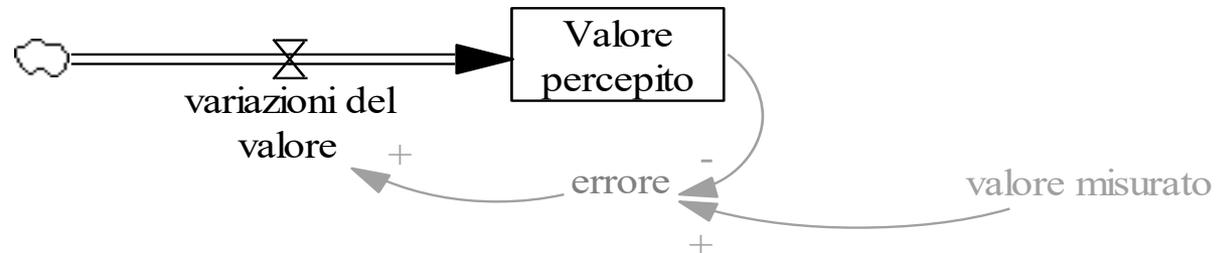
■ Material delay

- modella flussi “fisici”: tempi di fabbricazione, tempi di progetto, tempi “biologici”, tempi di servizio etc.
- vale la legge di **conservazione del flusso** (adatto a rappresentare dinamiche di **riempimento / svuotamento**)



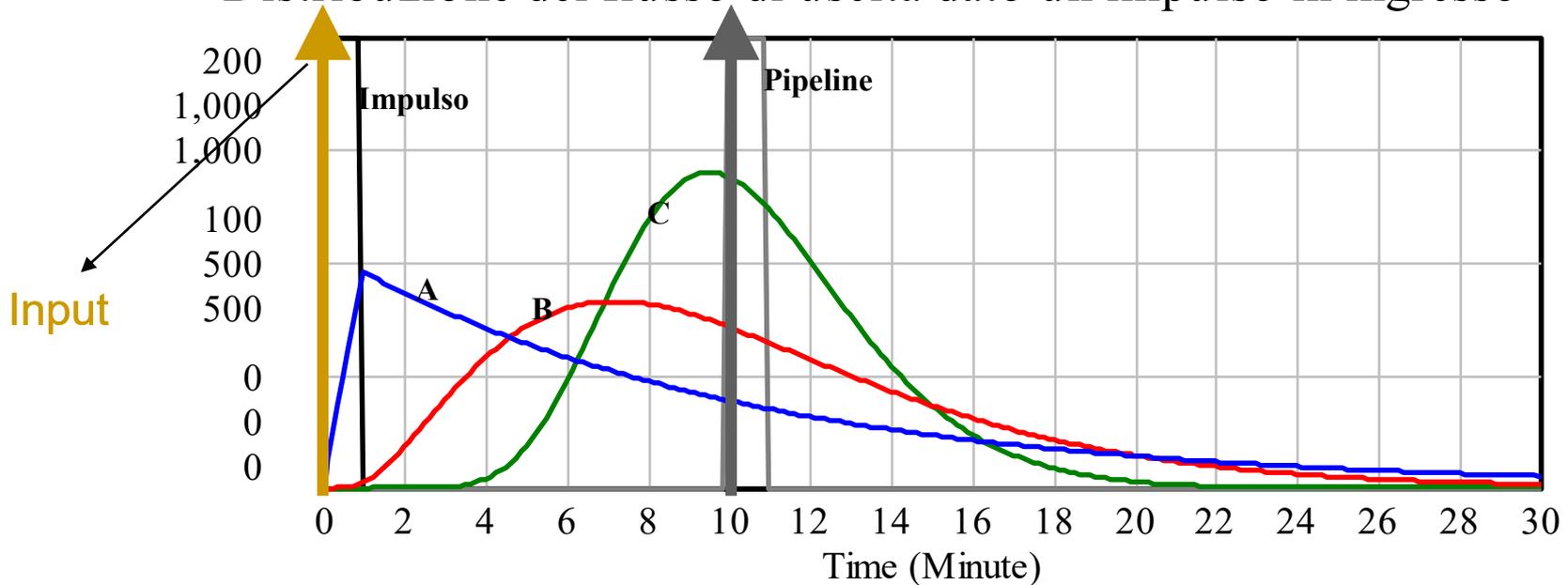
■ Information delay

- aggiustamenti graduali di “percezioni”: tempi di percezione, tempi di decisione ...
- la legge di conservazione di flusso può essere violata



Material delay: reference modes

Distribuzione del flusso di uscita dato un impulso in ingresso



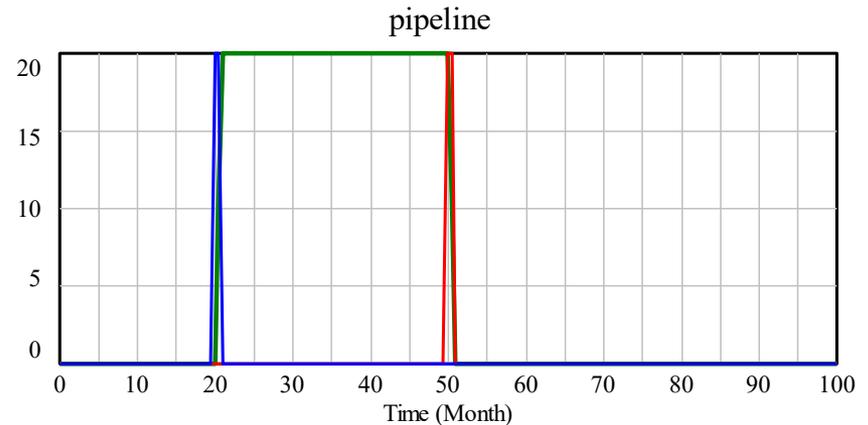
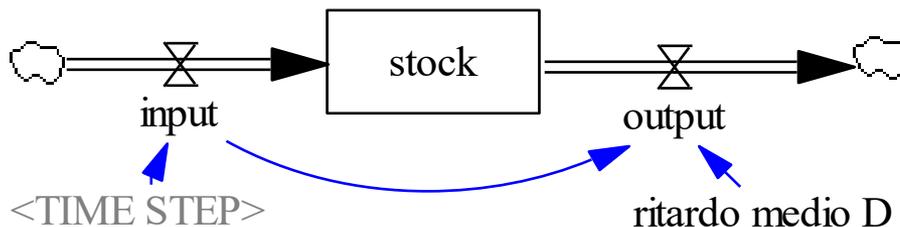
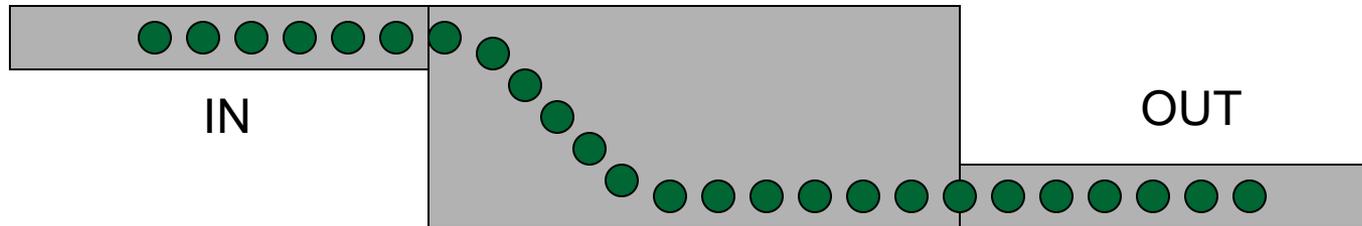
primo ordine : modello2
terzo ordine : modello2
decimo ordine : modello2
pipeline delay : modello2

- Stessa “area” (conservazione del flusso) e **stesso ritardo medio**

Nota: la system dynamics offre una **rappresentazione semplificata** dei sistemi di congestione (\neq simulazione discreta)

Material delay: pipeline

- Si mantiene l'ordine di ingresso (catena di montaggio)



- $output(t) = input(t - D)$

AnyLogic:

`output = delay (input , D)`

`output = delayMaterial (input , D , 0 , 0)`

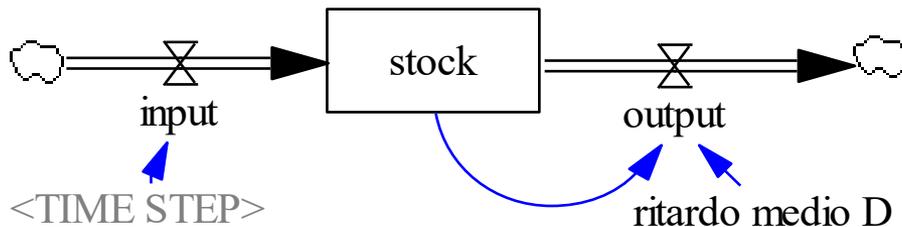
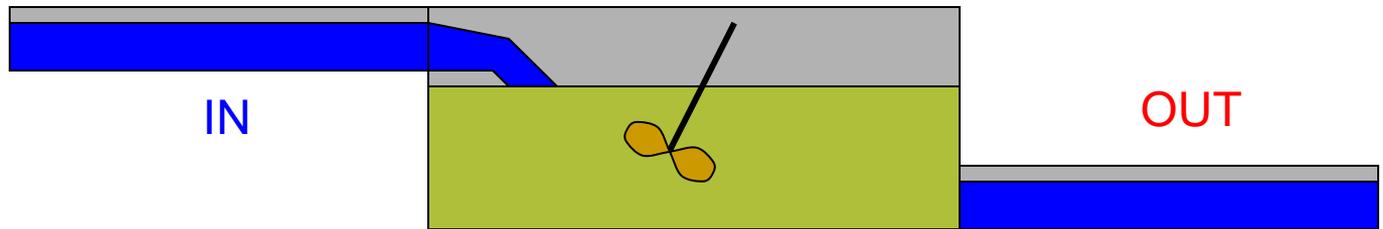
input : x
output : x
stock : x

`output=delay_fixed(input,d,0,0)`

vensim

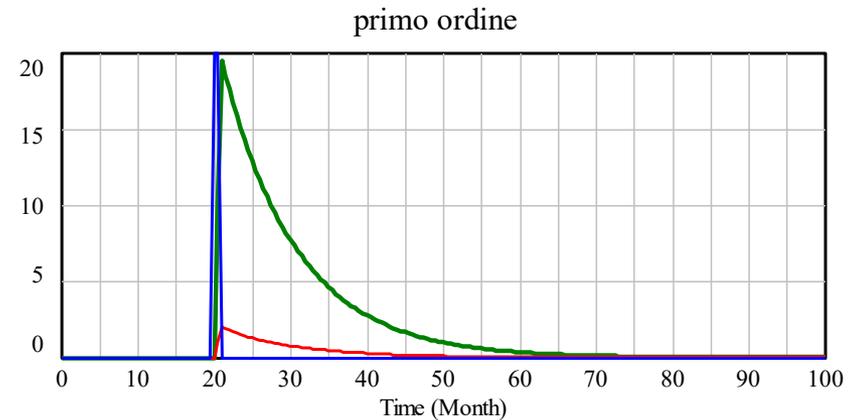
Material delay: primo ordine

- L'ordine di uscita è indipendente dall'arrivo: tutti hanno la **stessa probabilità** di uscire per primi (vasca con mixer)



- **$output(t) = stock(t) / D$**

Tasso di uscita direttamente proporzionale al livello dello stock e inversamente proporzionale al ritardo (se la coda alle poste aumenta, il direttore apre più sportelli ...)



input : x
output : x
stock : x

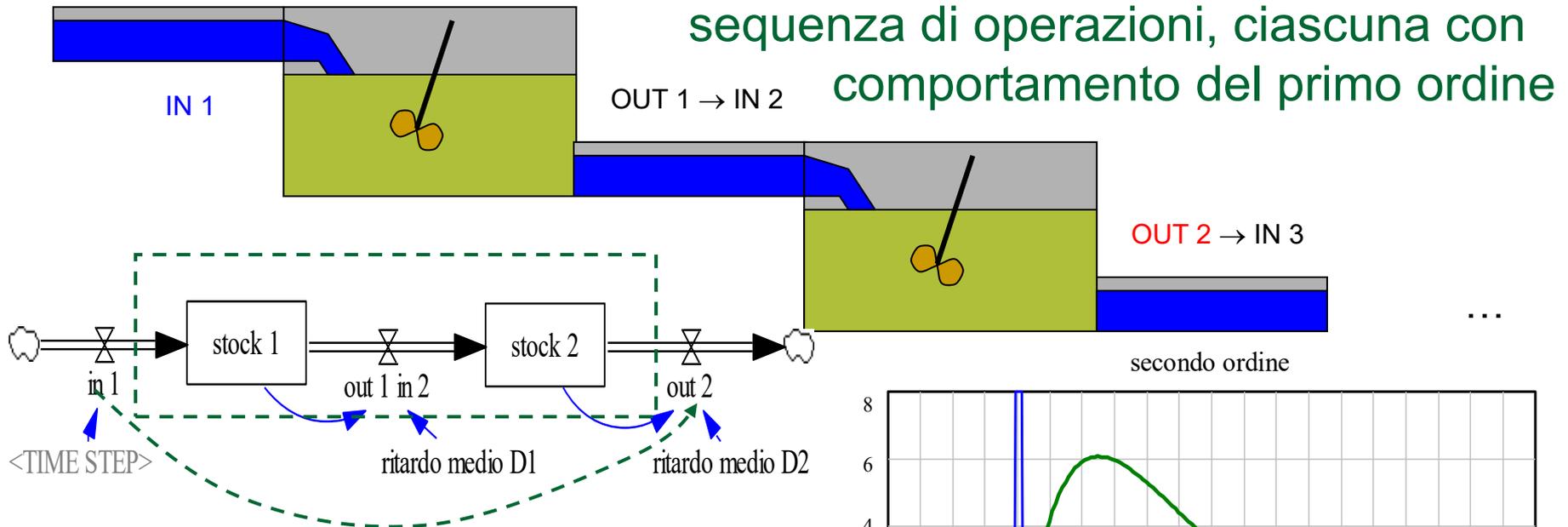
$$output = stock / D$$

$$output = delay1 (input , D)$$

$$output = delay_N(input, d, 0, 1)$$

Material delay: ordine superiore

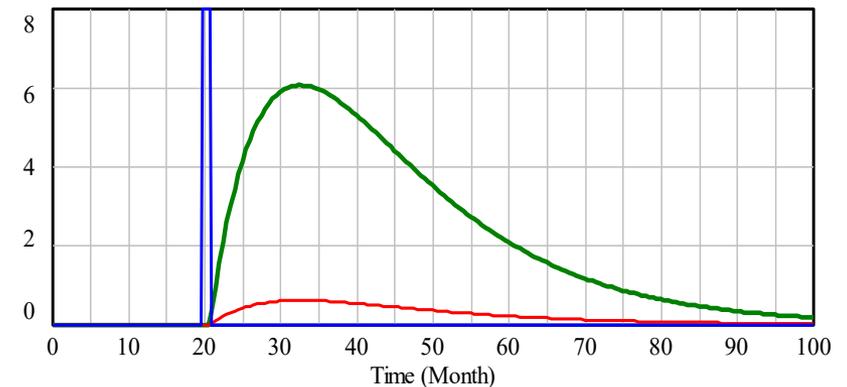
- Sistemi intermedi: l'ordine è mantenuto "a blocchi":



In Anylogic, solo **delay1** e **delay3** da combinare a cascata

Vensim:

`output=delay_N(in_1,d,0,1)`



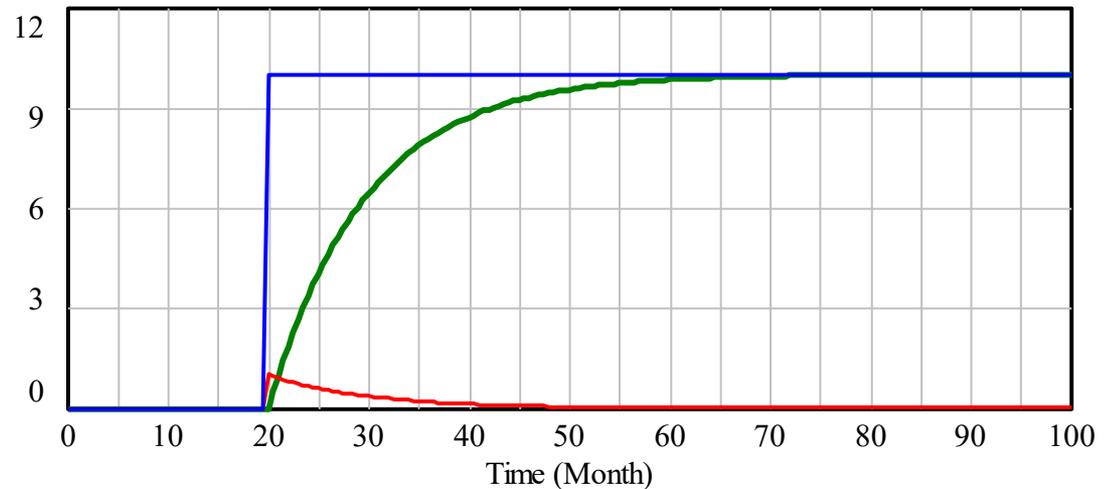
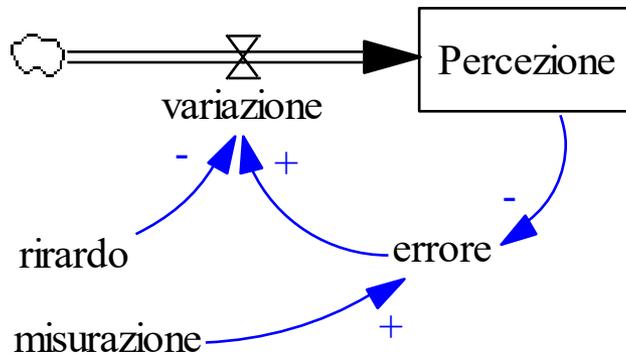
in 1 : x

out 2 : x

stock 2 : x

Information delay

- La **dinamica** è definita da un aggiustamento tra percezione e misurazione



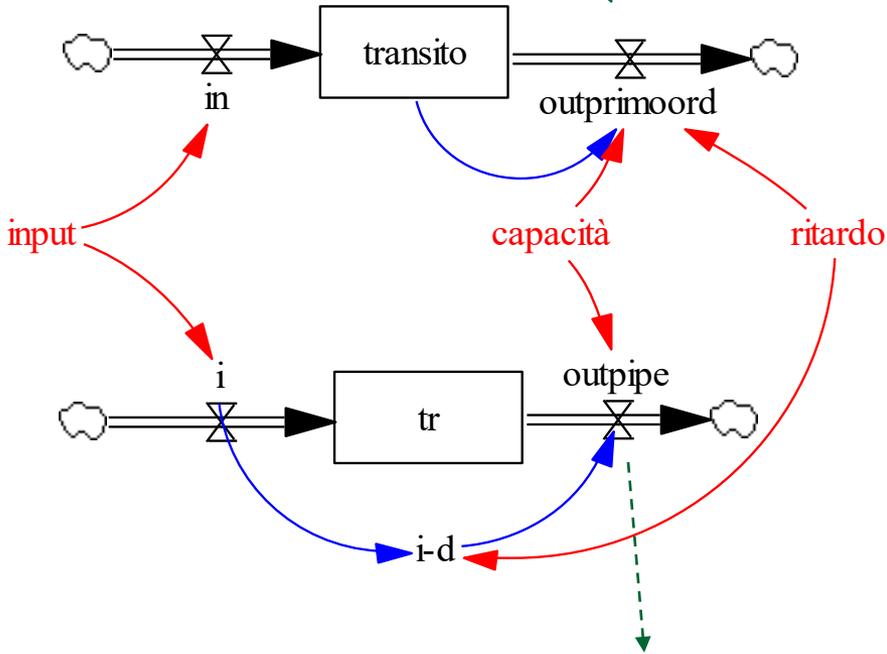
misurazione : x —————
variazione : x —————
Percezione : x —————

AnyLogic: non fornita, funzione **delayInformation** rappresenta dinamica diversa

Vensim: `Percezione = smooth_n (variazione, D , 0 , 1)`

Ritardi e capacità

$\text{MIN}(\text{capacità}, \text{transito/ritardo})$

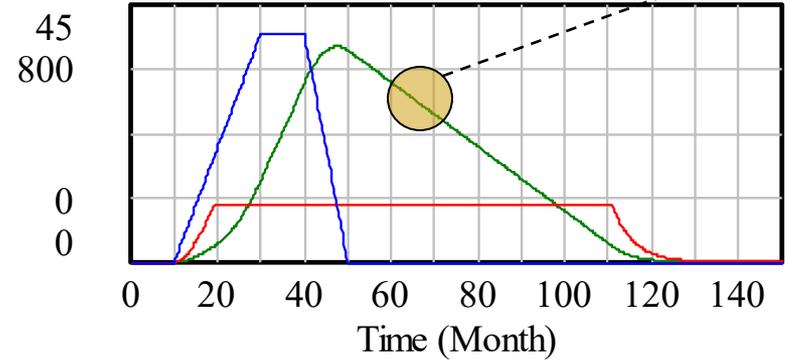


$\text{MIN}(\text{"i-d"}, \text{capacità})$

ritardi e capacità.mdl

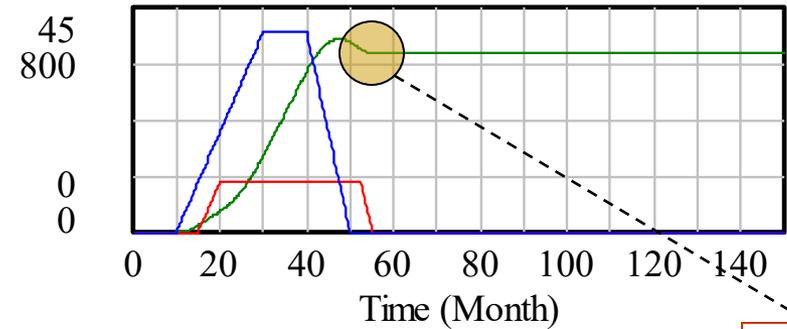
OK

Ritardo Primo Ordine e Capacità



in : x
outprimoord : x
transito : x

Ritardo Pipeline e Capacità



i : x
outpipe : x
tr : x

NO

I passi della simulazione

1. Poni $t = t_0$
2. Per ogni livello L , $L(t) = L(t_0)$ [noto]
3. Per ogni variabile ausiliaria V e flusso F calcola $V(t)$ e $F(t)$: in modo ricorsivo, utilizzando i valori di variabili e livelli al tempo t e fino alle variabili note a priori e/o ai livelli [ciclo \Rightarrow errore]
4. **Memorizza** tutti i risultati $L(t)$, $V(t)$ e $F(t)$ [dinamica]
5. **Integrazione**: per ogni livello L :
$$L(t + \Delta t) = L(t) + [F_{in}(t) - F_{out}(t)] \cdot \Delta t$$
6. Poni $t = t + \Delta t$
7. Se $t \leq T$ **torna** al passo 3
8. Fine: sono disponibili $L(t)$, $L(t)$ e $F(t)$, $t \in [t_0, T]$
(discretizzato)

Precisione dell'integrazione e Δt

- Δt deve essere *sufficientemente piccolo* per ridurre gli errori dell'approssimazione dell'integrale tramite discretizzazione:
accorgimenti pratici
 - scegliere una potenza negativa del 2 (errori numerici al calcolatore)
 - se la più piccola costante di ritardo è ε , $\Delta t \approx 1/3 \div 1/10 \varepsilon$
 - **dimezzare il valore di Δt finché i risultati della simulazione non diventano stabili**
- Attenzione:
 - ogni livello introduce un ritardo di Δt
 - Δt troppo piccoli possono portare ad errori numerici di arrotondamento e all'aumento dei tempi di calcolo (e delle dimensioni dei file che memorizzano i risultati)

Metodo di integrazione

- **Metodo di Eulero** (visto finora)
 - $L(t + \Delta t) = L(t) + [F_{in}(t) - F_{out}(t)] \cdot \Delta t$
- **Metodo Runge-Kutta** di ordine n
 - $L(t + \Delta t)$ dalla combinazione di n approssimazioni di $L(t + \Delta t/2)$
 - l'approssimazione corrisponde allo sviluppo di Taylor di ordine n
- In genere, il metodo di Eulero va bene (sono più grandi le approssimazioni introdotte dai dati e dalle semplificazioni del modello!)
- Il metodo di Runge-Kutta è utile per sistemi fisici con **oscillazioni persistenti**: gli errori di approssimazione di Eulero potrebbero essere amplificati (esempio: pendolo - [*pendulum.mdl*](#))
- In pratica
 - usare il metodo di Eulero dimensionando opportunamente Δt
 - verificare la robustezza dei risultati ripetendo la simulazione con stesso Δt e metodo di Runge-Kutta

Bussola

- Note introduttive
- Simulazione a eventi discreti: nozioni di base
- Simulazione a eventi discreti con AnyLogic
- Modellazione di sistemi dinamici
 - Relazioni causa-effetto e retroazioni
 - Modelli base di comportamento
- Rappresentazione di sistemi dinamici
 - Diagrammi causa-effetto
 - Reti flussi-livelli
- **Simulazione di sistemi dinamici con AnyLogic**
- Esempi notevoli di modellazione

Reti flussi-livello con AnyLogic

- Foglio sd.esercizi01.pdf
 - Esempio introduttivo
 - Modelli delle dinamiche-base
 - Modelli di diffusione

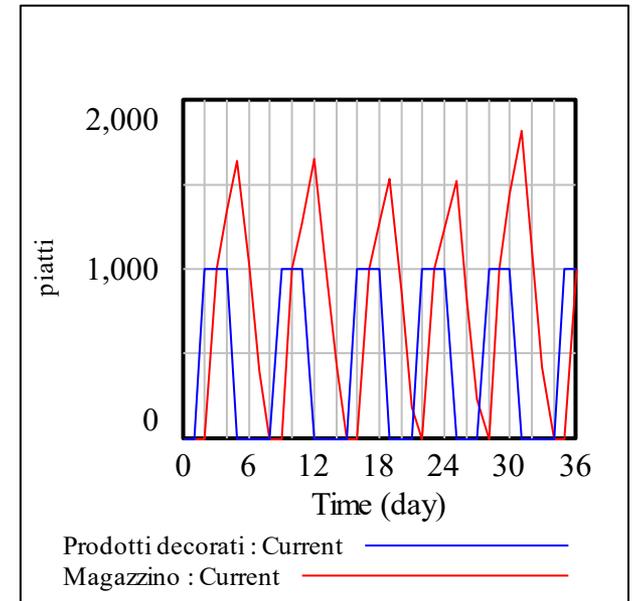
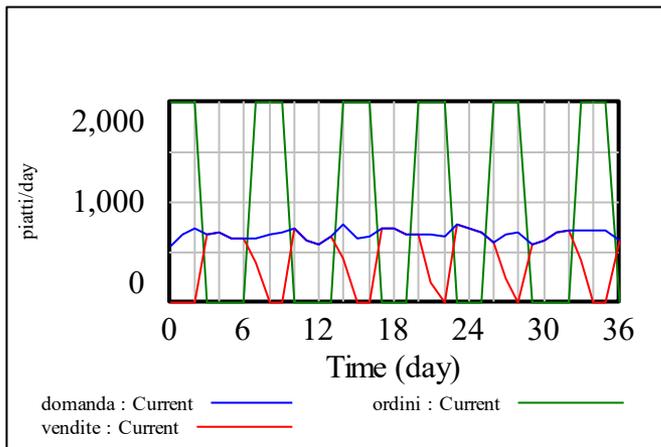
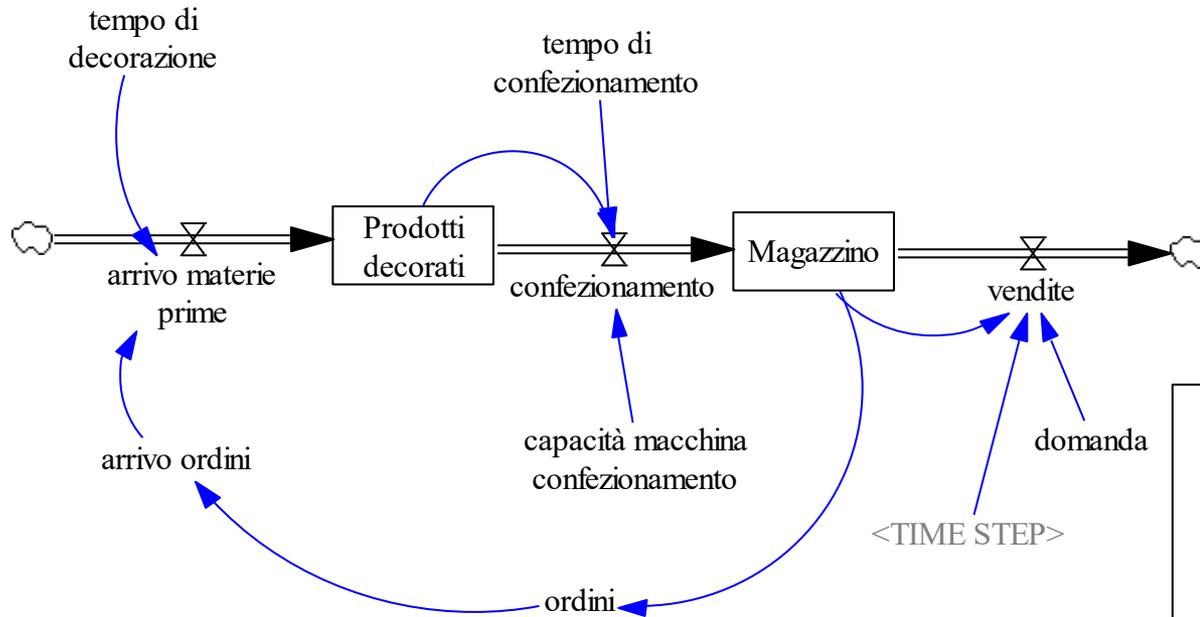
- Foglio sd.esercizi02.pdf: semplici applicazioni a
 - Modelli base dei ritardi
 - Dinamiche di svuotamento con capacità
 - sistemi di produzione (interesse alla dinamica più che agli elementi prodotti)
 - conto-corrente

Un processo produttivo (piatti – 1)

Un'azienda si occupa della fase di decorazione di piatti e della vendita del prodotto finito.

- Dopo la fase di decorazione, i prodotti decorati vengono confezionati e immagazzinati, in attesa di essere venduti.
- La fase di decorazione dura due giorni, e quella di confezionamento un giorno.
- La macchina per il confezionamento ha una capacità di 10000 piatti.
- Le vendite non sono regolari: il tasso di vendita giornaliero segue una distribuzione casuale di media 670 e deviazione standard di 55, con un massimo di 800 e un minimo di 500.
- La gestione del magazzino segue una politica a punto di riordino: se il livello del magazzino scende sotto le 1000 unità viene emesso un ordine di 2000 piatti grezzi. Il fornitore di piatti grezzi consegna dopo due giorni dall'emissione dell'ordine.

Un processo produttivo (piatti - 2)



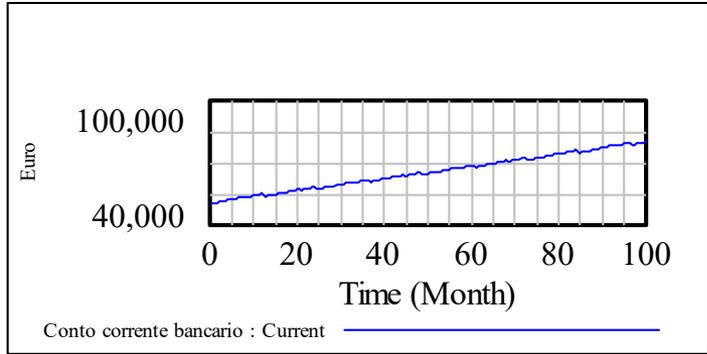
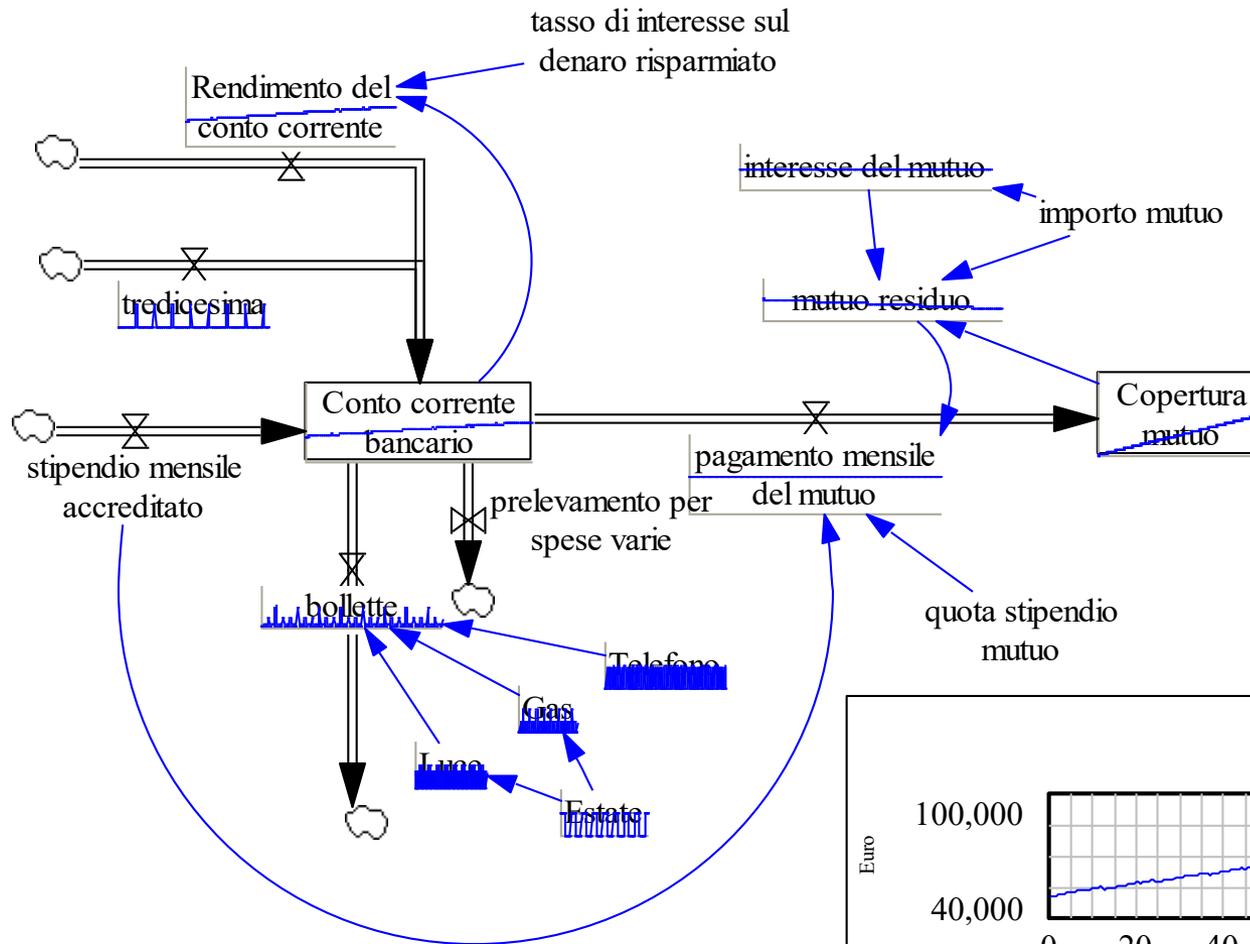
piatti.mdl

Conto corrente (c/c – 1)

Si vuole studiare la dinamica di un conto corrente, con particolare riferimento all'importo della rata di un mutuo.

- Il cliente accredita sul c/c lo stipendio mensile di 1500 €, e una tredicesima di 1300 €
- Le spese varie mensili ammontano a 500 €
- Il conto viene utilizzato per il pagamento delle bollette. La bolletta del gas arriva ogni 4 mesi e ammonta, in estate, a 300 € e, in inverno, a 700 €; la bolletta della luce arriva ogni due mesi e ammonta, in estate, a 40 € e, in inverno, a 70 €; la bolletta del telefono arriva ogni due mesi con un importo medio di 63 €
- Il c/c ha un rendimento dell'1 per mille mensile
- Il cliente ha contratto un mutuo di 100000 euro iniziali a tasso fisso e gli interessi da pagare complessivamente sono pari al 9% dell'importo. Il mutuo viene rimborsato con una rata mensile pari al 30% dello stipendio.

Conto corrente (c/c - 2)



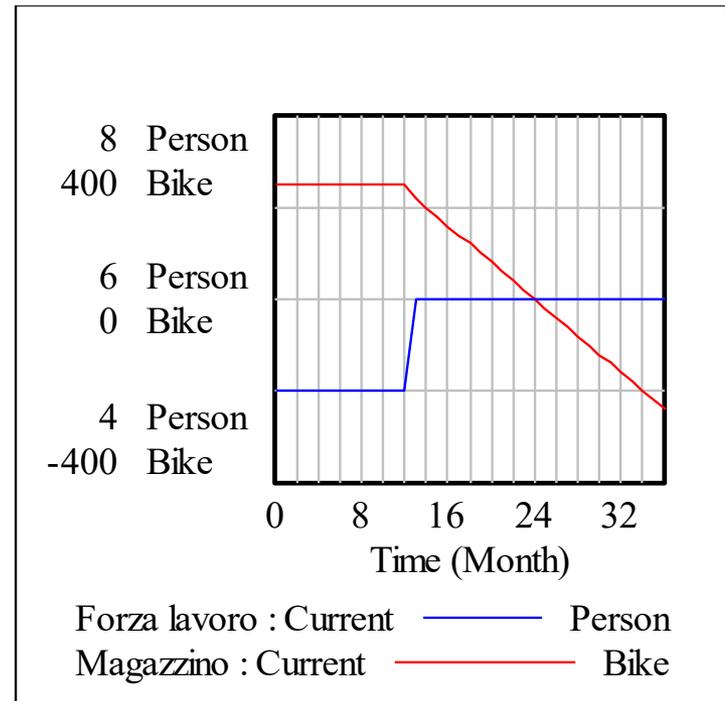
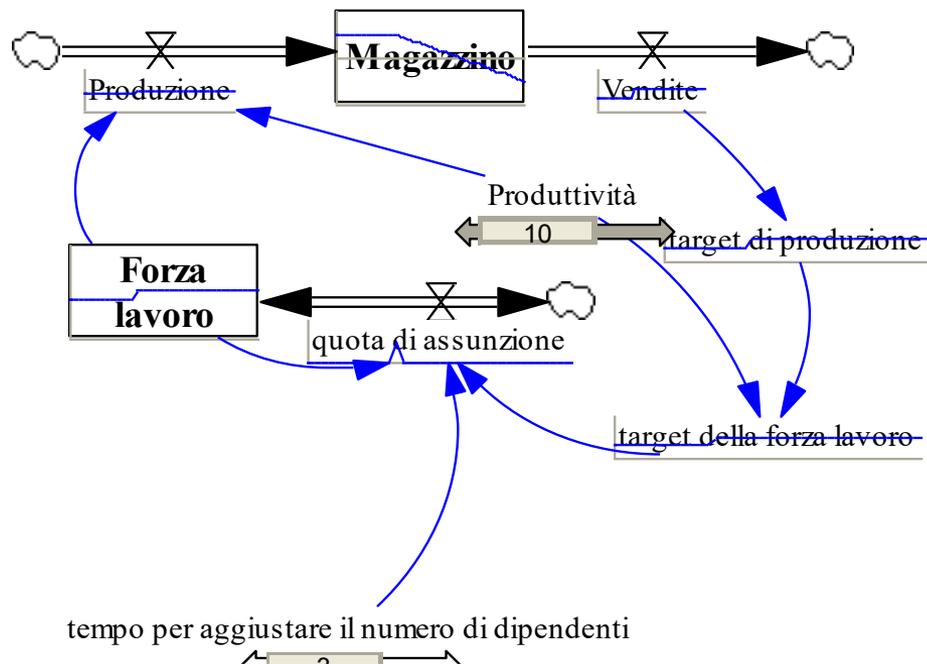
cc.mdl

Forza lavoro (bici – 1)

Un produttore di biciclette vuole analizzare la quantità di addetti nei prossimi due anni.

- Ogni addetto produce 10 bici al mese.
- Le vendite sono di 50 bici al mese, e si stima che, dal prossimo anno, cresceranno di 30 bici al mese.
- L'azienda è libera di assumere o licenziare operai. Il processo di reclutamento dura 3 mesi. Lo stesso periodo costituisce il preavviso per gli addetti da licenziare.

Forza lavoro (bici - 2)



[bici.mdl](#)

Bussola

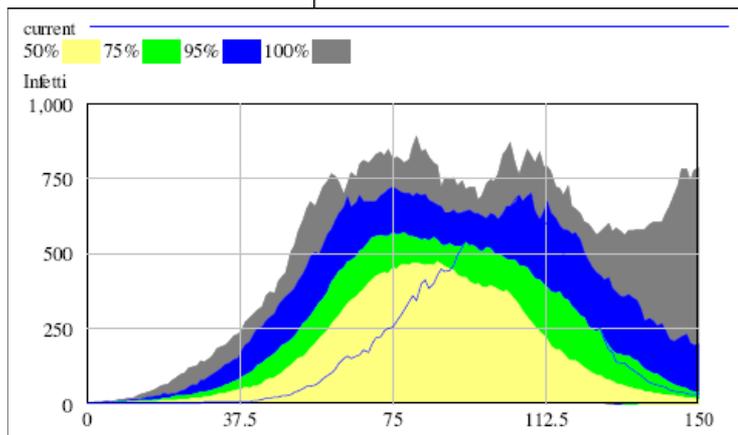
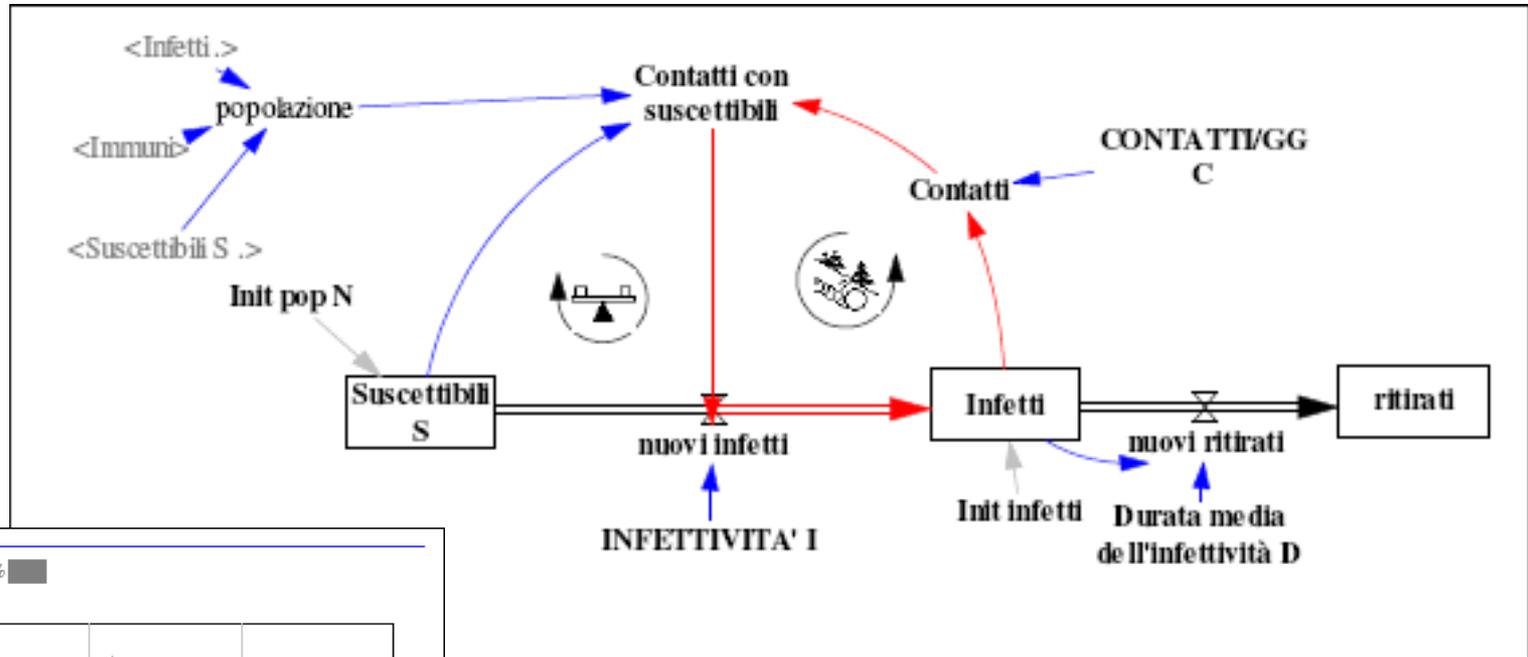
- Note introduttive
- Simulazione a eventi discreti: nozioni di base
- Simulazione a eventi discreti con AnyLogic
- Modellazione di sistemi dinamici
 - Relazioni causa-effetto e retroazioni
 - Modelli base di comportamento
- Rappresentazione di sistemi dinamici
 - Diagrammi causa-effetto
 - Reti flussi-livelli
- Simulazione di sistemi dinamici con AnyLogic
- **Esempi notevoli di modellazione**

Applicazioni notevoli

- Modelli di diffusione di virus (modelli SI, SIS, SIR: Susceptible – Infected – Retired)
- Modelli di impatto socio-economico (dinamica delle nascite e relative conseguenze)
- Modelli demografici (dinamica delle popolazioni, modelli preda-predatore)
- Modelli di sistemi di produzione (dinamica delle scorte)
- ...

Approfondimenti (SIR)

- Vedi [SIR.PDF](#) Nota: **SI** = Susceptible Infected
SIR = Susceptible Infected Retired (o Resolved)



Altre applicazioni (cura ing. R. Berchi)

- Gestione progetti ([gestione progetti.pdf](#))
- Modelli di mercato ([modelli mercato.pdf](#))
- Modelli di produzione ([produzione.pdf](#))
- Supply chain ([supply chain.pdf](#))
- Catene di invecchiamento - aging chains
- Modelli macroeconomici
 - Modelli di sviluppo ([innovazione.pdf](#))
 - Modelli di Solow
 - Modelli di Samuelson
- Modelli microeconomici
 - andamento economico/finanziario di un'azienda (dinamica dei bilanci)
- Ecologia
 - Ecosistemi (es. preda-predatore)
 - Diffusione di inquinanti ([appunti prof. Gallo](#) – 6.3.2)
 - Inquinamento atmosferico ed effetto serra ([appunti prof. Gallo](#) – 6.3.3)
- **etc. etc. etc.**