Are Processes Occurrents, Continuants, Both, or Neither?

Antony Galton

College of Engineering, Mathematics, and Physical Sciences
University of Exeter, UK

The word “process” is used in several different ways

Process I

“the topic-neutral counterpart of [Vendler’s] activity”

(Mourelatos 1981)

In this sense a process is

- homogeneous (i.e., homeomerous and additive)
- imperfective (i.e., open-ended, atelic)

Process II

A completable routine comprising a structured sequence of actions or events.

In this sense, “process” is *close* to “event”, whereas in sense I it is typically *contrasted* with “event”.

The process of

$\ldots$ *making a pot of tea*
$\ldots$ *checking in for a flight*
$\ldots$ *assembling a model from a kit*
$\ldots$ *applying for a new passport*

This is the kind of process that is typically referred to in expressions of the form “I am in the process of Xing”

Process III

> “Processes are repeatable behaviours whose occurrences cause continuants to undergo change”
>
> Özgövde and Grüninger, 2010

A process, in this sense, is an abstract pattern of behaviour that can be realised in the form of specific occurrences (events) located in space and time.

PSL — Process Specification Language (Grüninger *et al*)

Process IV

In Computer Science, the term “process” is often used to refer to some program code, rather than the behaviour it generates.

Process algebras / process calculi:

- Dynamic Logic (Pratt, 1976)
- Communicating Sequential Processes (Hoare, 1978)
- π-Calculus (Milner, 1999)

PROBLEM

To develop an account of processes which takes into account all of the above.

Generic Basic Process

A **generic basic process** is a homogeneous, open-ended behaviour which may be **enacted** by an **agent** (or set of agents) over a period of time.

("Homogeneous" means relative to some chosen level of granularity.)

Generic basic processes are denoted by simple verbs:

run
sing
eat
whistle
flow

Specific Basic Process

A **specific basic process** is obtained from a generic basic process by specifying an agent (or set of agents) for it.

Specific basic processes are denoted by verb phrases consisting of a simple verb together with a subject and, optionally, a *non-delimiting* object (typically either a mass terms or an indefinite plural):

John run
Mary sing
Mary sing Schubert
Bill eat
Bill eat apple(s)
the kettle whistle
the river flow

Subjectless Processes

For processes without a (logical) subject, the distinction between generic and specific collapses:

(it) rain
(it) become dark

Delimited Basic Process

A **delimited basic process** is the result of assigning to a basic process (either generic or specific) a limiting qualification which determines beginning and end points — thus negating the open-endedness of the undelimited basic process.

The limiting qualification may be

- temporal ("for an hour")
- spatial ("for a mile")
- material ("the/an [apple]")

Delimited Basic Processes: Examples

Delimited Generic Basic Processes:

run a mile
sing for an hour
sing *Fairest Isle*
eat an apple

Delimited Specific Basic Processes:

John run a mile
Mary sing for an hour
Mary sing *Fairest Isle*
Bill eat an apple

A delimited specific basic process is an example of an **event type**.

Realisations of Basic Processes

All of the entities considered so far are abstract, i.e., not located in space and time, neither continuants nor occurrents.

They can be regarded as schemas, templates, or types, to which concrete spatio-temporal entities can be assigned as **realisations**.

Realisations of processes are of two kinds,

- **continuant realisations** — states
- **occurrent realisations** — events

Continuant Realisations of Basic Processes

An **continuant realisation** of a basic process is a **state** which holds by virtue of a specific basic process being enacted at a moment of time:

John running at t
Mary singing at t

(Note: These are non-delimited specific basic processes — see below for continuant realisations of delimited specific basic processes)

A continuant realisation can can also be said to persist (or endure) over an interval:

John running throughout $[t_1, t_2]$

Occurrent Realisations of Basic Processes

An **occurrent realisation** of a basic process is an **event** which occurs by virtue of the enactment of a delimited specific basic process over an interval:

John run a mile over $[t_1, t_2]$
Mary sing *Fairest Isle* over $[t_1, t_2]$
Mary sing for an hour over $[t, t + \text{one hour}]$

Occurrent realisations are **event tokens**; they are instances of the event types represented by the delimited processes of which they are realisations.

N.B., the linguistic expression of an occurrent realisation does not have to mention the interval explicitly.

Continuant realisations of delimited basic processes

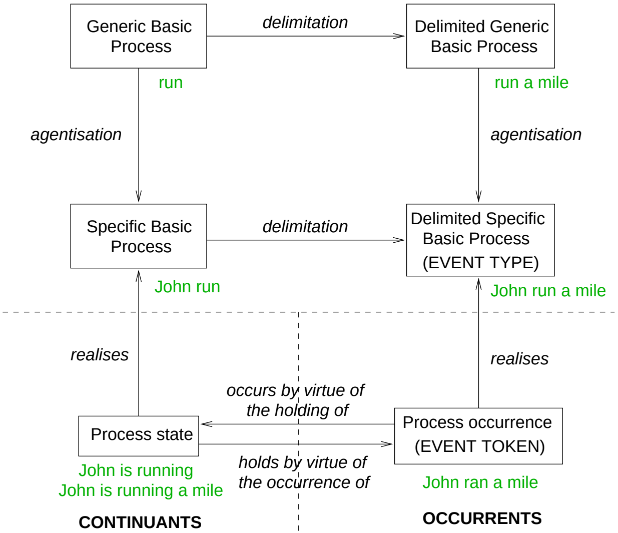
Although in some sense inherently occurrent-like, delimited basic processes can have continuant realisations:

John running a mile at t
Mary singing for an hour over $[t, t +$ five minutes$]$

These continuant realisations may or may not form part of a complete enactment (i.e., an occurrent realisation) of the process concerned.

- The "Imperfective Paradox"
- Ambiguity of "Mary was singing for an hour"

A B S T R A C T



Generic Basic Process

delimitation

Delimited Generic Basic Process

run

run a mile

agentisation

agentisation

Specific Basic Process

delimitation

Delimited Specific Basic Process (EVENT TYPE)

John run

John run a mile

realises

realises

occurs by virtue of the holding of

Process state

Process occurrence (EVENT TOKEN)

holds by virtue of the occurrence of

John is running
John is running a mile

John ran a mile

CONTINUANTS

OCCURRENTS

C O N C R E T E

A realisation of a delimited specific basic process consists of a suitable process state holding for as long as is required for the delimitation condition to be satisfied:

- John run a mile — John run holds until John has covered a distance of one mile.
- Mary eat an apple — Mary eat holds until the apple is consumed.

This does not work for a type II process, i.e., "a completable routine comprising a structured sequence of actions or events":

- John make a pot of tea — *John make holds until a pot of tea is ready.

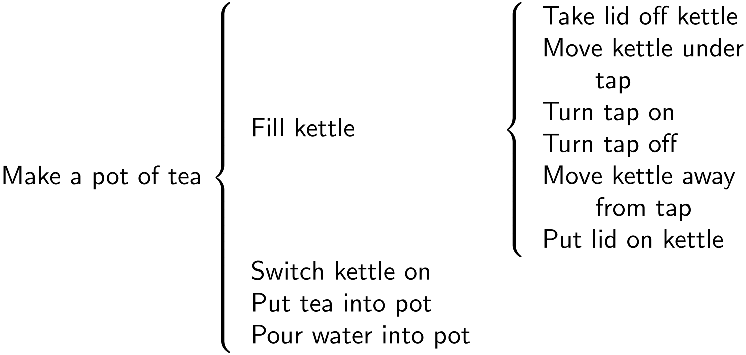
There is no generic basic process make to play the role of run and eat in the previous examples.

Non-basic processes

John make a pot of tea is a **non-basic process**: it is not possible to characterise what you have to do to make a pot of tea under a single description — rather, a realisation of the process consists of **structured sequence of heterogeneous actions**.

The constituent actions may be realisations of basic processes — though we may have to go through a process of **stepwise refinement** to uncover the basic constituents.

Stepwise refinement of "make a pot of tea" (incomplete)



- Make a pot of tea
  - Fill kettle
    - Take lid off kettle
    - Move kettle under tap
    - Turn tap on
    - Turn tap off
    - Move kettle away from tap
    - Put lid on kettle
  - Switch kettle on
  - Put tea into pot
  - Pour water into pot

Analysis of delimited basic processes

Move the kettle under the tap:

- Generic basic process: Move the kettle
- Delimitation: until the kettle is under the tap

Turn the tap on:

- Generic basic process: Turn the tap
- Delimitation: until the tap is on

Pour the water into the pot:

- Generic basic process: Pour water [from the kettle]
- Delimitation: until the [right quantity of] water is in the pot

Hypothesis

Every action in a structured routine can eventually be broken down into delimited generic basic processes.

Question

What are the possible modes of combination of delimited generic basic processes?

Modes of combination of delimited basic processes

Computer programming:

A_1*; *A_2 *(i.e, do* A_1 *followed by* A_2*)*
repeat A *until* ϕ
while ϕ *do* A
if ϕ *do* A_1 *else do* A_2

The "repeat" command applies to discrete actions — since the execution of a program consists of a sequence of discrete steps. If continuous action is possible, we also need:

continue A *until* ϕ*.*

Move the kettle under the tap $=$ *continue* move the kettle
until the kettle is under the tap

Other formalisms: Dynamic logic; π-Calculus; etc

Composition Operations

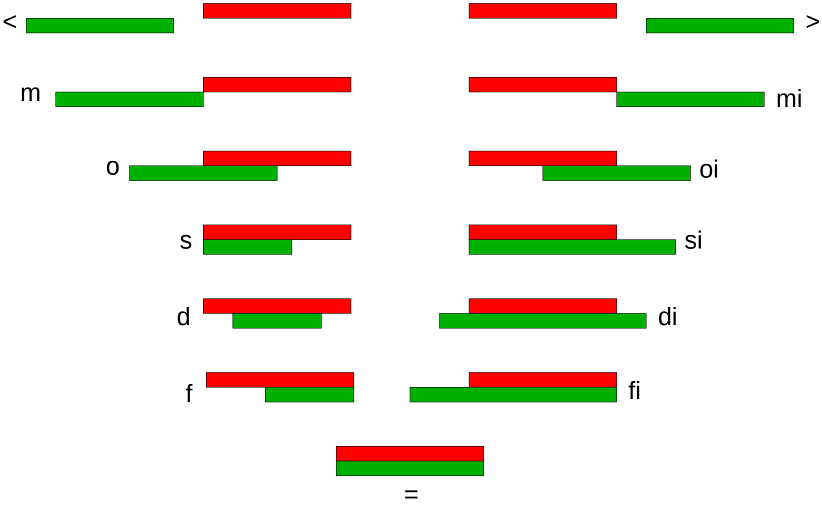
General scheme: A composition of event-type A with event type B is an event type each of whose occurrences consists of an occurrence of A and an occurrence of B in a specified temporal relationship.

Traditionally, one distinguishes **sequential** and **parallel** composition, but we need a more fine-grained set of distinctions.

The temporal relationships may be purely qualitative, expressed by means of the Interval Calculus relations (Allen, 1984) — e.g., *before*, *overlap*, *starts*, *during*, or disjunctions of such relations.

Or they may be in part quantitative — e.g., *not more than three seconds after*.

Interval Calculus relations



Examples of composition using Allen relations

- **Opening a door with latch.** You have to disengage the latch, and while it is still disengaged, push the door; once the door has started to open it is not necessary to keep the latch in the disengaged position.

  Keep latch disengaged {o,fi,di} Push door

- **Playing a stopped note on a violin**. With a finger of the left hand you have to stop the string at the position appropriate to the note you want to play; once the string is stopped, with your right hand you have to draw the bow across the string.

  Stop string {=,si,fi,di} Draw bow across string

What can we compose?

The example

Stop string $\{=,\mathrm{si},\mathrm{fi},\mathrm{di}\}$ Draw bow across string

shows a composition of delimited generic processes, resulting in a delimited generic process.

If the player is specified we have had a composition of delimited specific processes resulting in a delimited specific process:

Mary stop string $\{=,\mathrm{si},\mathrm{fi},\mathrm{di}\}$ Mary draw bow across string

Realisation of a composition

A composite delimited specific process might have the form

$$dsp_1 \; S \; dsp_2$$

where S is a set of interval relations.

A realisation of this process on the interval i consists of a realisation of dsp_1 on an interval i_1 together with a realisation of dsp_2 on an interval i_2, where

- i_1 stands stands to i_2 in one of the relations in S,
- i is the convex hull of $i_1 \cup i_2$.

Mixed compositions

A delimited generic process can be composed with a delimited specific process, resulting in a delimited generic process.

Example. The central portion of fill the kettle is the composition

$$\text{turn tap on}\ \{\text{m}\} \left[\begin{array}{c}\text{desired amount of water}\\ \text{enter kettle}\end{array}\right] \{\text{m}\}\ \text{turn tap off}$$

where the second component is specific, the first and third generic.

An “agentised” version is John fill the kettle:

$$\left[\begin{array}{c}\text{John turn}\\ \text{tap on}\end{array}\right] \{\text{m}\} \left[\begin{array}{c}\text{desired amount of water}\\ \text{enter kettle}\end{array}\right] \{\text{m}\} \left[\begin{array}{c}\text{John turn}\\ \text{tap off}\end{array}\right]$$

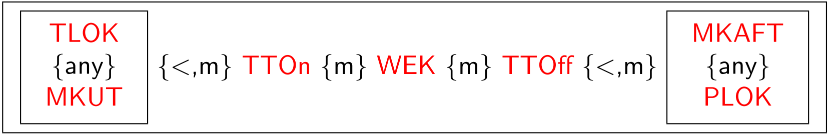
Compositional analysis of “make a pot of tea”

Key:		
	FK	fill kettle
	BK	boil kettle
	TLOP	take lid off pot
	PLOP	put lid on pot
	PTIP	put tea in pot
	PWIP	pour water into pot

$$\boxed{\begin{array}{c}\boxed{\text{FK } \{<,\text{m}\} \text{ BK}} \\ \{>,\text{mi},\text{oi},\text{f},=,\text{fi},\text{si},\text{di}\} \\ \boxed{\text{TLOP } \{<\} \text{ PTIP}}\end{array}} \; \{\text{m}\} \text{ PWIP } \{<\} \text{ PLOP}$$

Refinement of “fill kettle”

Key:	TLOK	take lid off kettle
	MKUT	move kettle under tap
	TTOn	turn tap on
	WEK	desired amount of water enters kettle
	TTOff	turn tap off
	MKAFT	move kettle away from tap
	PLOK	put lid on kettle



TLOK
{any}
MKUT

{<,m} TTOn {m} WEK {m} TTOff {<,m}

MKAFT
{any}
PLOK

Delimitations

The delimited process “Continue $\langle \textit{process} \rangle$ until $\langle \textit{state} \rangle$” is analysed as a composition of the form “*occurrent realisation of* non-*delimited process* finished-by *inception of state*”.

The inception of state S is the event-type $ingr(S)$ defined by

> $ingr(S)$ occurs at t iff there are intervals i and j such that
> – i meets j at t
> – $-S$ holds on i
> – S holds on j

For non-delimited process P and state S, “Continue P until S” is analysed as the delimited process P{fi}ingr(S).

move the kettle under the tap =
move the kettle {fi} *ingr*(the kettle is under the tap)

Conclusions I

- A **process** is an abstract "template" that can be realised both as an occurrent and as a continuant.
- A **generic process** specifies an activity without attributing it to an actor; when a generic process is attributed to an actor, the result is a **specific process**.
- But some processes are "**subjectless**", and for these the distinction between generic and specific collapses.
- A **continuant realisation** of a process is a state.
- An **occurrent realisation** of a process is an event.

Conclusions II

- Primarily, a process realised as a continuant is **non-delimited**, and a process realised as an occurrent is **delimited**.
- A **continuant realisation of a delimited process** is a realisation of the corresponding non-delimited process; in describing it as delimited, one is describing it as *actually* or *possibly* giving rise to an occurrent realisation of the delimited process.
- An **occurrent realisation of a non-delimited process** is a realisation of a delimited process derived from it; in describing it as non-delimited, one is simply omitting to mention any delimitation.
- **Composite processes** are built up from **basic processes** using **composition operations**.