Creativity and Innovation in Software Engineering Teaching

Hussein Zedan

We acknowledge

- **the support of the SEDiLia project (Bulgaria) and**
- **the collaboration with colleagues in the STRL and DMU (UK)**

September 1, 2009

STRL

Issues and Problems

- Conceptual advances have always been the deriving force behind progress. This, in turns, relies on *creativity* and the ability to continue the production of new insights and novel ideas.
- This is even more so for the *teaching* processes and techniques of any subject.
- However, teaching is *hard*, specially of any science/engineering subjects. It requires input from variety of different disciplines: Education Theorists, Psychologists, Sociologists, etc.
- Software engineering educators must teach students to *think* creatively and to discover innovative solutions to *real* problems.
- This has become a crucial requirement as we are advancing/progressing towards e/m-Learning modes of education.

STRL

- Current teaching techniques (of software engineering) *do not* encourage creativity. Indeed, in many situations, they do hinder creativity!
- In a phenomena-based teaching, we
  1. start with explaining a phenomenon, with the help of oversimplified (toy) examples.
  2. This is followed by testing students using equally oversimplified (toy) problems similar to those used in the explanation phase!
  3. If still not clear AND there is time, simplify the toy examples even more and go to (1)!
- Both phases are done within the same context.
- Lack of time, stringent educational policies, standard is dropping , etc. are often attributed to the adoption of the above.

STRL

- Fred Marlin has argued for more realistic educational context for software engineering. He claims that teaching should be *interactive* and *collaborative*, moving away from the oversimplified toy examples.
- Chuan-Hoo Tan has also argued that giving students experience developing and delivering large-scale systems under time constraints and shifting deadlines can better prepare them for future challenges.

- Many projects (at undergraduate and postgraduate levels) lack both a *fun/excitement* factor to engage students and the *practical realism* of engineering projects that include other computer science disciplines such as networks and HCI.
- Some have introduced courses (even degrees) on *games* development.
- Ian Parberry and his colleagues explored the use of game programming with art students , arguing that such an approach creates the opportunity for diverse communities of students to collaborate on joint projects.

STRL

- Similar experience was reported by Ming-Hsin Tsai on applying game design in the education of art and design students. Their students have made complete games and not just oversimplified excercises or simple walk-through scenes.
- The rethinking CS101 Project (www.cs101.org) claims that most introductory programming courses - which typically teach computation as sequential problem solving - are outdated. Rather, such courses should emphasis interaction among processes.

STRL

Limitations

- Whilst these approaches/recommendations for creative teaching can help to some extent, they suffer some drawbacks:
  1. We talk of creative teaching yet we have no commonly agreed understanding of the notion of creativity . In addition to a lack of a scientific underpinning to support their claims.
  2. Creativity has a temporal dimension. What might have been creative then may not be creative now. Any approach must enable students to cope with *paradigm* shifts.
  3. Creative teaching must have an individualistic dimension. The suggested approaches have an implicit belief that it is one-fits-all solution! Approaches for creative teaching should be tailored to the need of individual student.

STRL

Computational Thinking - A General principle

- Computational thinking builds on the power and limits of computing processes, whether they are executed by human or by a machine.
- Computational models and methods give us the courage to solve problems and design systems that no one of us would been capable of tackling alone.
- Computational thinking is a fundamental skill for everyone. To reading , writing and arithmetic, we should add computational thinking to every child's analytical ability.
- Computational thinking involves solving problems, designing systems and understanding human behaviour, by drawing on the concepts fundamental to computer science.
  Computational thinking includes a range of mental tools that reflect the breadth of the field of computer science.

STRL

Computational Thinking: Characteristics

- Computational thinking has the following characteristics:
  1. Conceptualising, not programming/coding.
  2. Fundamental, not rote skill.
  3. A way that humans, not computers, think.
  4. Complements and combines mathematical and engineering thinking.
  5. Ideas, not artifacts.
  6. For everyone, everywhere.
- These are all well known principles. However, they lack the how:

  How the adoption of the above leads to the enhancement of creativity.

STRL

Research Questions

- Can the discovery and analysis of creative processes enhance the creative teaching techniques?
- Do mechanisms of the creative processes cross the boundaries of disciplines?
- Are there optimal conditions that may enhance the creativity in Software Engineering education?
- How does cooperation/collaboration affect creativity?

To shed some light on the above questions, we provide a *unifying* framework within which the creative processes (and hence creativity) can be understood and analysed. The unification here is in the sense that the proposed framework is discipline-independent.

STRL

1. Biological sciences

gttgaggggtgttgagggcggagaaatgcaagtttcattacaaaagttaacgtaacaaa
aatctggtagaagtgagttttggatagtaaaataagtttcgaactctggcacctttcaat
tttgtcgcactctccttgtttttgaca

atgcaatcatatgcttctgctatgttaagcgtattcaacagcgatgattacagtccagct
gtgcaagagaatattcccgctctccggagaagctcttccttcctttgcactgaaagctgt
aactctaagtatcagtgtgaaacgggagaaaacagtaaaggcaacgtccaggatagagtg
aagcgacccatgaacgcattcatcgtgtggtctcgcgatcagaggcgcaagatggctcta
gagaatcccagaatgcgaaactcagagatcagcaagcagctgggataccagtggaaaatg
cttactgaagccgaaaaatggccattcttccaggaggcacagaaattacaggccatgcac
agagagaaatacccgaattataagtatcgacctcgtcggaaggcgaagatgctgccgaag
aattgcagtttgcttcccgcagatcccgcttcggtactctgcagcgaagtgcaactggac
aacaggttgtacagggatgactgtacgaaagccacacactcaagaatggagcaccagcta
ggccacttaccgcccatcaacgcagccagctcaccgcagcaacgggaccgctacagccac
tggacaaagctgtag

gacaatcgggtaacattg
gctacaaagacctacctagatgctccttttacgataacttacagccctcactttcttat
gtttagtttcaatattgttttctttctctggctaataaaggccttattcatttca

2. Automated reasoning

- Reasoning is the process of systematically drawing conclusions from a number of given facts by applying a series of steps.
- There is no doubt that those steps in any mathematical proof, by their brevity and unexpected turns, may strike its reader as being very ingenious constructions.
- Dijkstra has observed that: ' [...] many of those steps that may seem surprising, at first sight, are, in fact, (almost) dictated, as they are the only (or by far the simplest) transformation that will enable to exploit one of the givens that has to be taken into account.'
- In this connection, he recommended that ' [...] we maintain as fine grained a bookkeeping as possible of what of the givens we have used: what has not been used yet often indicates the direction in which the proof should be completed.'

STRL

Interactivity vs Legacy Creation

- One of the major *outcomes* is the construction process of *Creativity Maps* which provide the necessary basis for studying the creative processes and hence enhancing our understanding of the creative phenomenon.
- The creativity maps can be built in an incremental, interactive and a non-intrusive fashion (i.e., as the creation is being developed).
  For legacy creations, the maps can also be constructed a *posteriori* but only if adequate information were available.
  E.g. Mozart vs. Beethoven.

STRL

Forces and Challenges

- Understanding and analysing *creativity* and the *creative processes* are hard.
- At its core, creativity are both subjective and domain-oriented .
- Both *novelty* and *value* have often been attributed to creativity. These attributes are very hard to evaluate.
- Adding to the challenges is that creativity has a temporal dimension.

STRL

Models

- Some dismiss the notion that creativity can be described as a sequence of steps in a model. But while such views are strongly held, they are in the minority.
- In business, where models are used for quality improvement, strategic planning, re-engineering, and so on, are well-positioned to deal with this apparent controversy. Whilst models may appear to be useful and helpful in guiding our efforts, they should not be used too rigidly for that is perceived to constrain creativity.
- On the other hand, even if we deviate substantially from a model in a given situation, this does not render the model useless. It is also important to understand that we should not be too rigid about when one step of a model ends and the next begins.

STRL

Models - cont.

- Anecdotal descriptions have been also used to identify processes that are considered creative. Many discoveries, specially in the sciences were linked to a sudden realisation or unexplained divine intervention associated with what is known as the *AHA! response*
- For example, Newton's falling apple (which has been since disputed!), Archimedes' "Eureka" moment in a bath or Mendeleev's dream are well known examples.

STRL

- Would it have been profitable if we had some records of what Isaac Newton was thinking, together with, his moods "before" and "after" the fall of that apple?
- Equally, it would have been extremely interesting and useful to have known Archimedes' moods "before" and "after" he went to that famous bath! And surely, knowing
- Mendeleev' states "before" and "after" sleeping might have shed some lights on his discovery.

The "before" and "after" states of mind are very powerful mechanisms to analyse creativity and the creative processes.

STRL

Models - cont.

- Most of existing models have a common characteristic: they depend on a balance between analytical and synthetic thinking and usually describe the creative process as a sequence of phases that alternate between these states.
- The implied theory behind older models is that the creative thinking is a subconscious process that cannot be directed, and that creative and analytical thinking are complementary. This is known as the AHA! response/factor.
- Modern models however tend to imply purposeful generation of new ideas, under the direct control of the thinker.

STRL

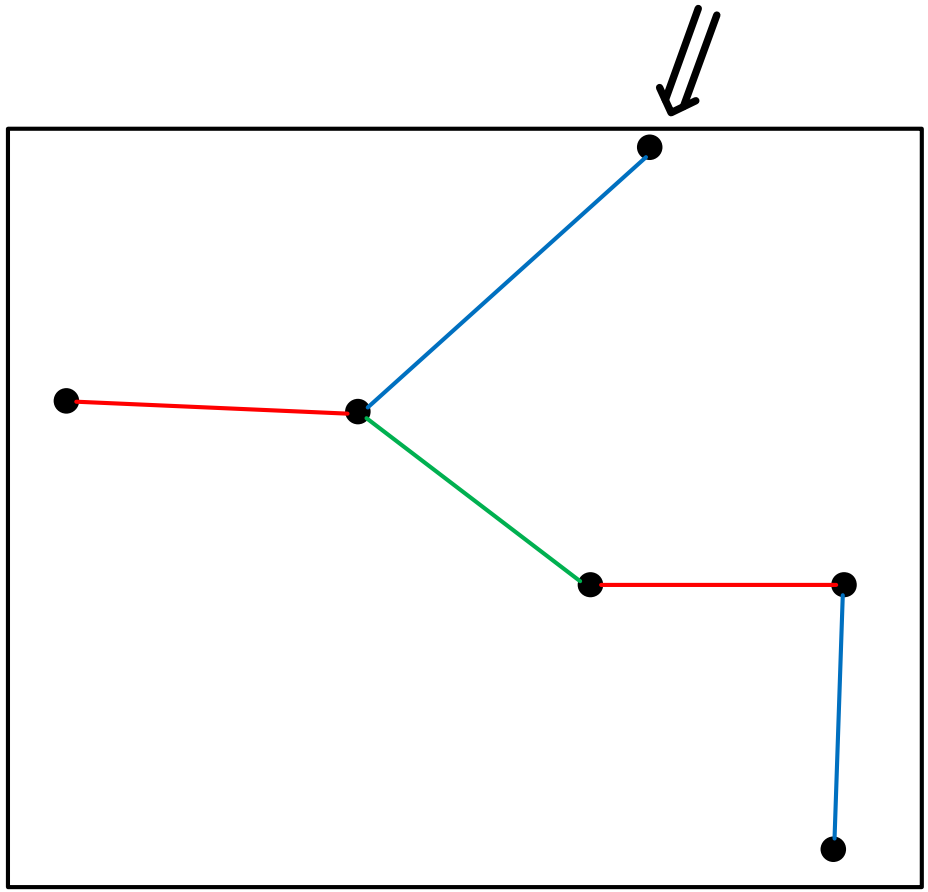
Axioms of Creativity

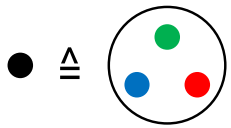
A0. Creativity is identified only by its product.

A1. The value of creativity is determined only by the society it receives it.

A2. Creativity is an emergent phenomenon.

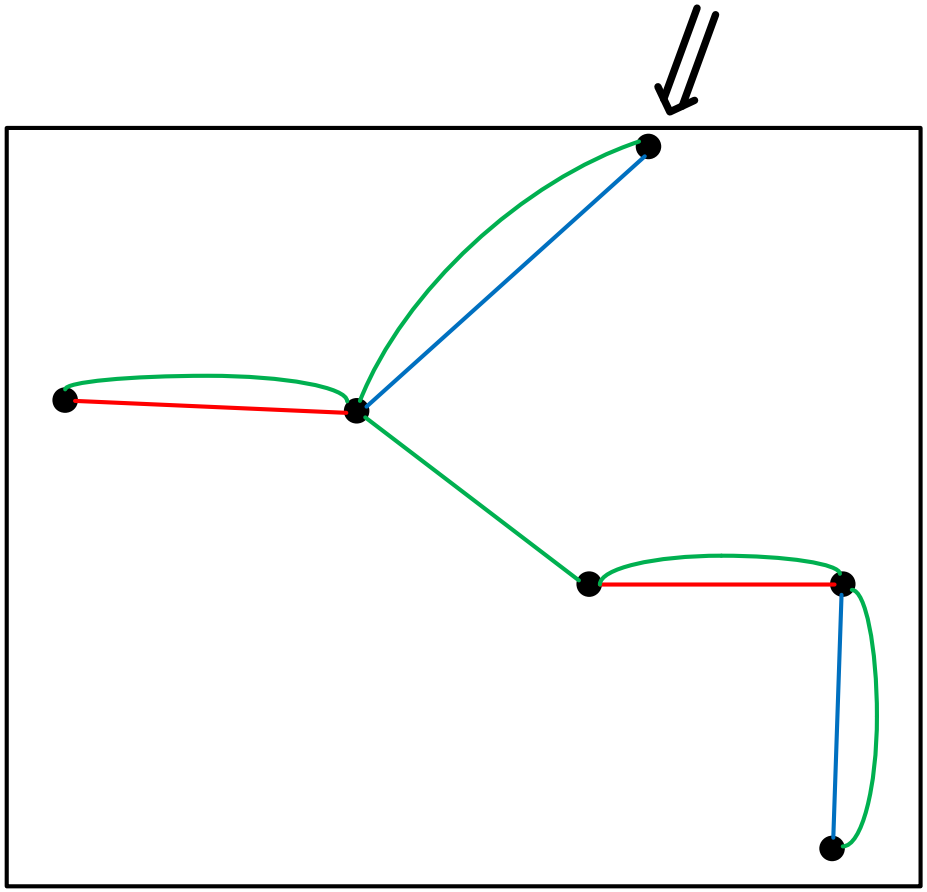
STRL

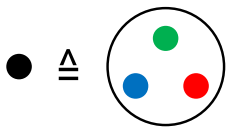




Multiple Views
(Universe!)

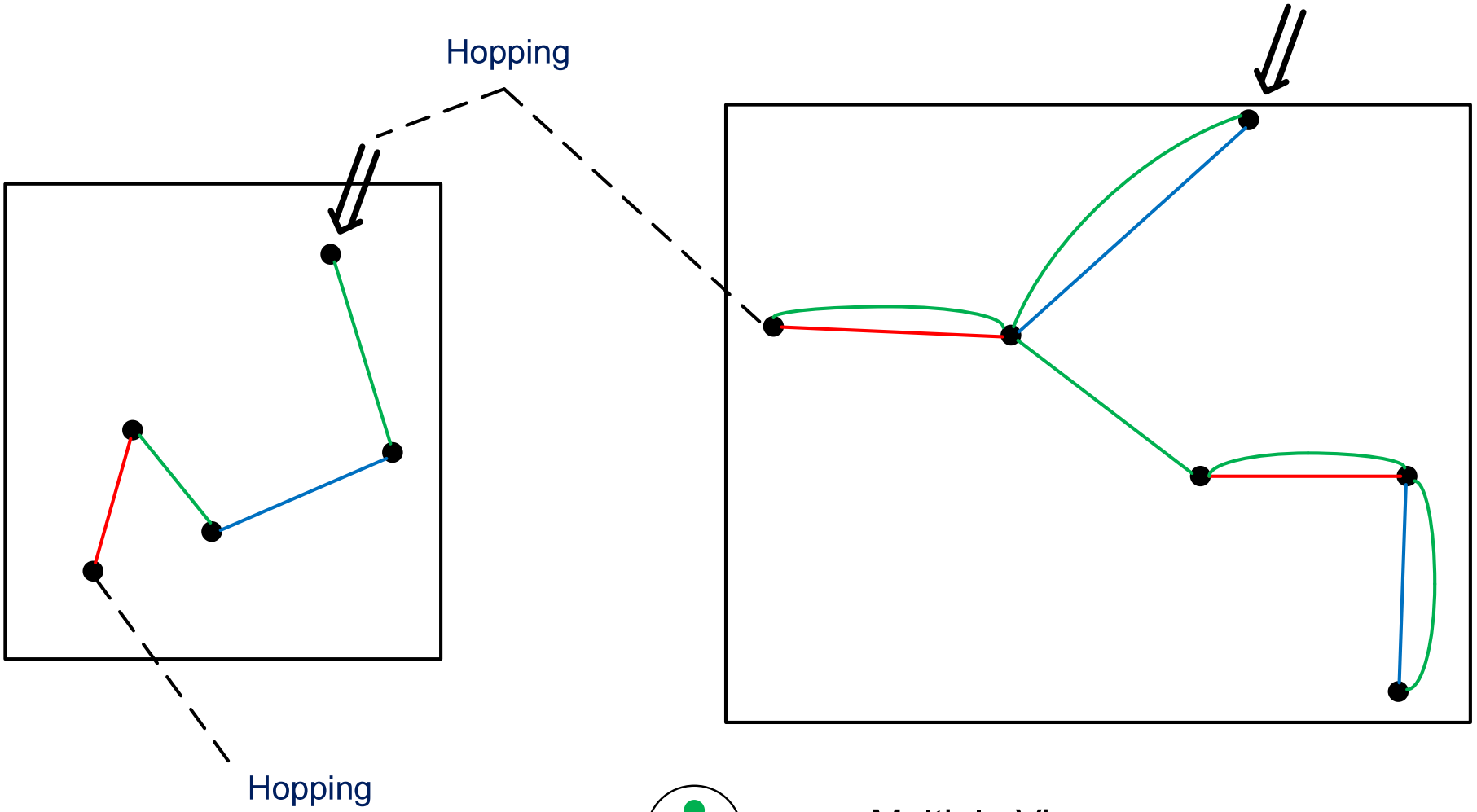
STRL



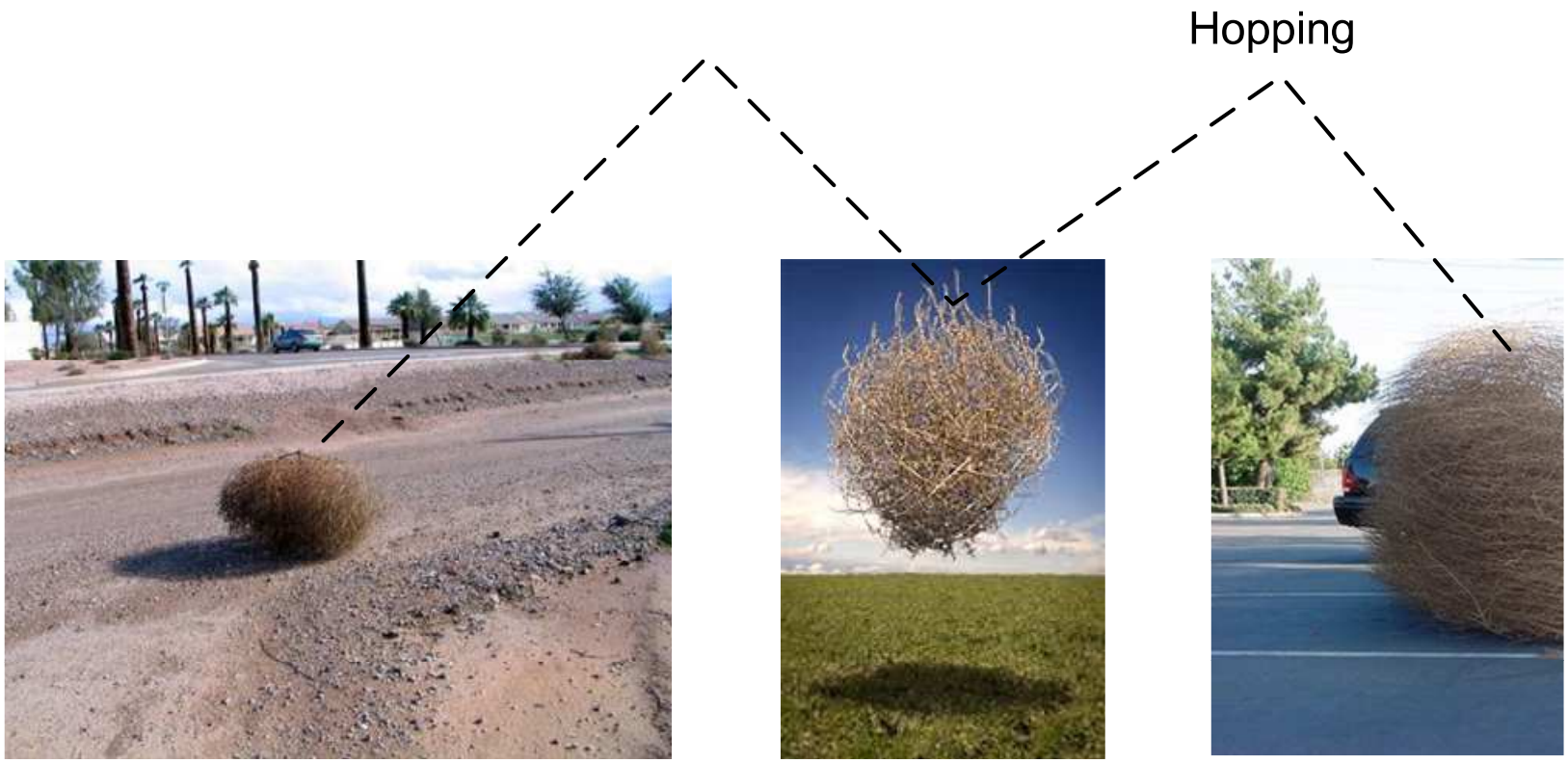


Multiple Views
(Universe!)

STRL

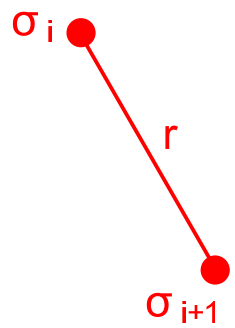


STRL


Hopping

Transitions & Transition Systems

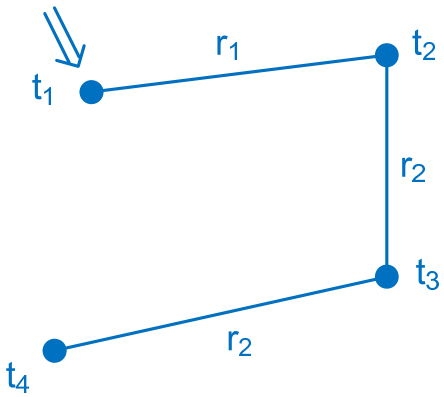
- Let Σ be a set of states ,
  $\sigma_i \in \Sigma$,
  $\sigma_i : Var \rightarrow Val$
- A transition r (sometimes denoted by $\rightarrow_r$)
  is a relation
  $r \in \Sigma x \Sigma$
- Let R be a set $s \cdot t$. $r \in R$, i.e.,
  $R \in P(\Sigma x \Sigma)$



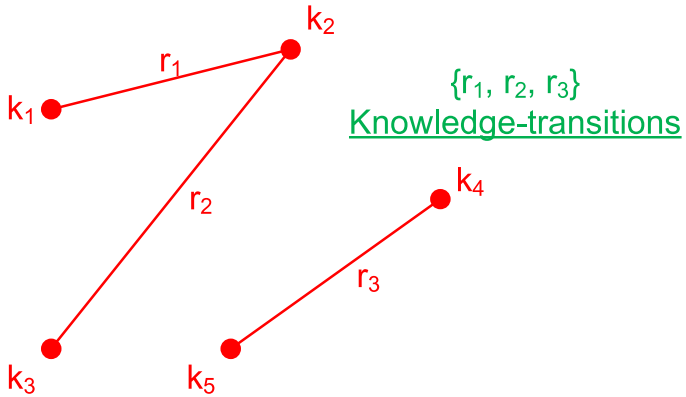
STRL

- (Σ, R) is called a Transition System

Clock



Knowledge



$Var = Clock$
$Val = \mathbb{N}$

$\rightarrow \; t_i \; r_1 \; t_{i+1} \mathrel{\widehat{=}} (t_{i+1} = t_i + 1)$

$\rightarrow \; t_i \; r_2 \; t_{i+1} \mathrel{\widehat{=}} (t_{i+1} \geq t_i)$

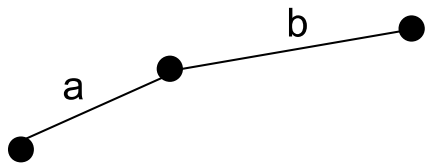
r_1, r_2 is a Clock-transition

Var = Knowledge
Val = $k_1, k_2, k_3, ...$

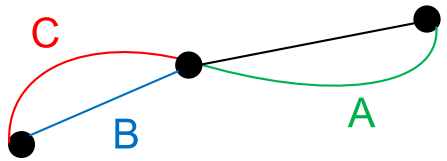
- $k_2 \; r_2 \; k_3 \mathrel{\widehat{=}} (k_2 = k_3)$
- $k_2 \; r_1 \; k_1 \mathrel{\widehat{=}} (k_1 > k_2)$
- $k_4 \; r_3 \; k_5 \mathrel{\widehat{=}} (k_5 \leq k_4)$

STRL

Classic transition systems use fixed relation between states.
Actions are used to label transition.



In our Generalised Transition System, there may be more than one relation between states.
Actions vary from one relation to another.



STRL

Operators & Functions

- For a given transition

$$\sigma_i \; r \; \sigma_j$$

St: $\Sigma \times \Sigma \rightarrow \Sigma$

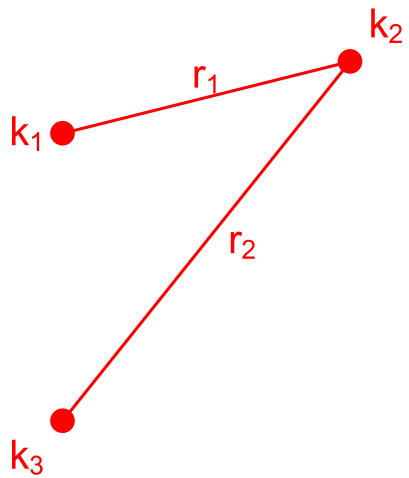
$\sigma_i \; r \; \sigma_j \mapsto \sigma_i$

Et: $\Sigma \times \Sigma \rightarrow \Sigma$

$\sigma_i \; r \; \sigma_j \mapsto \sigma_j$

Knowledge

$$St(r_1) = St(r_2) = k_2$$
$$Et(r_2) = k_3$$
$$Et(r_1) = k_1$$



STRL

- Obviously

$$St(R) \widehat{=} \bigcup_{r \in R} St(r)$$

$$Et(R) \widehat{=} \bigcup_{r \in R} Et(r)$$

STRL

ϵ-transition

For any Σ, an ϵ-transition is defined as

$$\forall s \in \Sigma : \quad s \in s.$$

ANY-transition

For any Σ, an Any-transition is defined as

$$ANY \mathrel{\widehat{=}} \Sigma \times \Sigma$$

(all possible transitions)

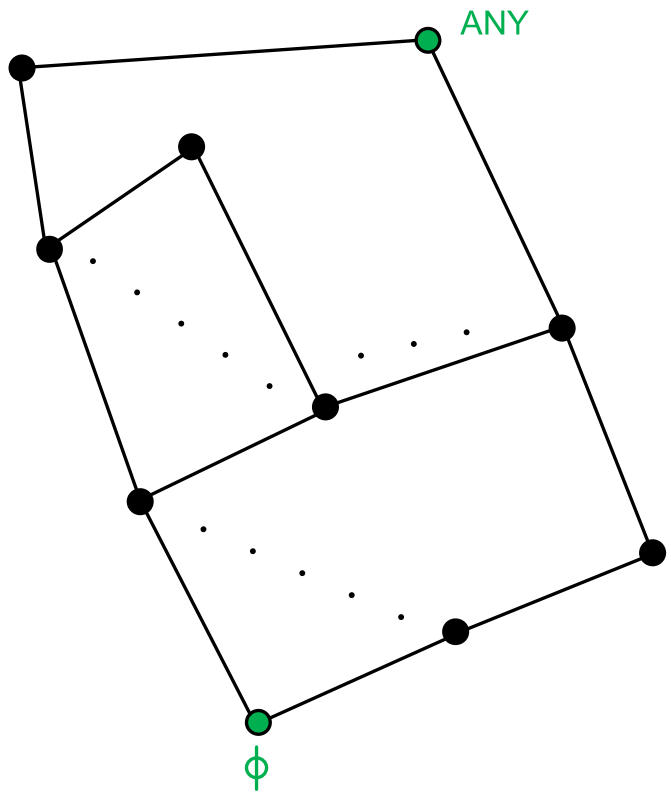
STRL

NULL-transition

For any Σ, a NULL-transition, ϕ, is

$$\forall \sigma, r \in \Sigma : \quad \neg(\sigma \phi r)$$

- ANY is maximal relation
- ϕ is mininmal relation.

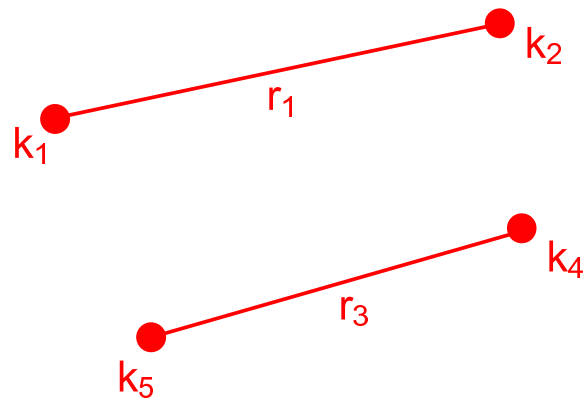


STRL

OR:

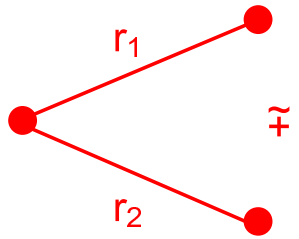
$$\rightarrow_{r_1} + \rightarrow_{r_2} \widehat{=} \{(\sigma_i, \sigma_j) : (\sigma_i \rightarrow_{r_1} \sigma_j) \vee (\sigma_i \rightarrow_{r_2} \sigma_j)\}$$

$$\rightarrow_{r_1} + \rightarrow_{r_2} = \{(k_2, k_1), (k_4, k_5)\}$$



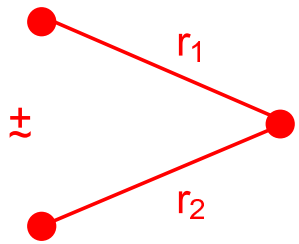
- $\rightarrow_{r_1} \tilde{+} \rightarrow_{r_2} \widehat{=} \begin{cases} \rightarrow_{r_1} + \rightarrow_{r_2}, & St(\rightarrow_{r_1}) = St(\rightarrow_{r_2}) \wedge \\ & Et(\rightarrow_{r_1}) \neq Et(\rightarrow_{r_2}) \\ \phi & otherwise \end{cases}$

Branching



- $\rightarrow_{r_1} \tilde{+} \rightarrow_{r_2} \widehat{=} \begin{cases} \rightarrow_{r_1} + \rightarrow_{r_2}, & Et(\rightarrow_{r_1}) = Et(\rightarrow_{r_2}) \wedge \\ & St(\rightarrow_{r_1}) \neq St(\rightarrow_{r_2}) \\ \phi & otherwise \end{cases}$

Merge



Sequential

For any Σ and $\rightarrow_{r_i}$,

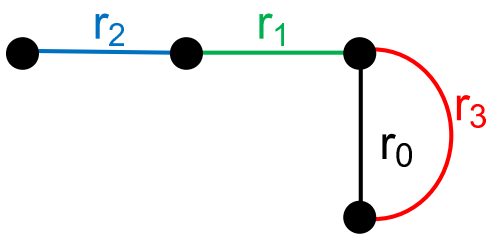
$\sigma_1 \bullet \rightarrow_{r_i} \bullet\sigma_2 \; ; \; \sigma_3 \bullet \rightarrow_{r_j} \bullet\sigma_4 \mathrel{\widehat{=}}$

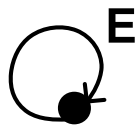
$$\begin{cases} \sigma_1 \bullet \rightarrow \bullet\sigma \rightarrow \bullet\sigma_4 & , \sigma = \sigma_2 = \sigma_4 \\ \mathit{null} & , \mathit{otherwise} \end{cases}$$

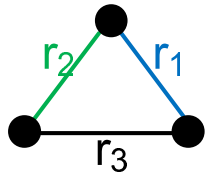
Sequential-Path $\rightarrow^*$

For any Σ, $\rightarrow^*$ is defined as:

$$\rightarrow^* \widehat{=} \in + \rightarrow j \rightarrow^*$$







Creativity Laws I

- $NULL+ \rightarrow_r = \rightarrow_r = \rightarrow_r + NULL$
- $\epsilon; \rightarrow_r = \rightarrow_r = \rightarrow_r; \epsilon$
- $ANY+ \rightarrow_r = ANY = \rightarrow_r + ANY$

- $NULL; \rightarrow_r = NULL = \rightarrow_r; NULL$ (finite)
- $NULL^* = \epsilon$

- $\epsilon^* = \epsilon$
- $ANY^* = T \qquad (T = (\Sigma \times \Sigma)^*)$

Creativity Laws I - Cont.

- $\rightarrow_r ; (\rightarrow_t + \rightarrow_w) = (\rightarrow_r ; \rightarrow_t) + (\rightarrow_r ; \rightarrow_w)$
- $\rightarrow_r + (\rightarrow_t + \rightarrow_w) = (\rightarrow_r + \rightarrow_t) + \rightarrow_w$
- $\rightarrow_r ; (\rightarrow_t ; \rightarrow_w) = (\rightarrow_r ; \rightarrow_t) ; \rightarrow_w$

STRL

Creativity Laws I - Cont.

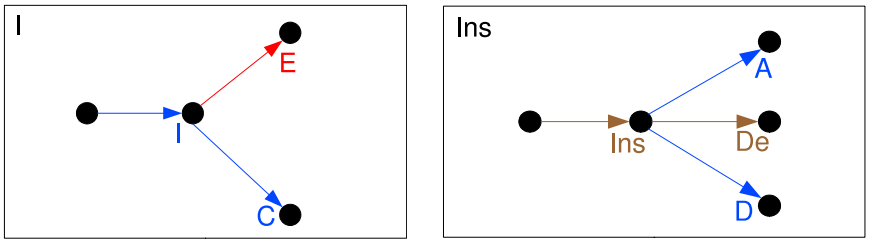
- $\rightarrow_r^* = \epsilon + \rightarrow_r ; \rightarrow_r^*$
  $= \epsilon + \rightarrow^* ; \rightarrow_r$
- $(\rightarrow_t + \rightarrow_r)^* = \rightarrow_t^* ; (\rightarrow_r ; \rightarrow_t^*)*$
- $\rightarrow_{r_1} + \rightarrow_{r_1} = \rightarrow_{r_1}$

PLUS Many More!

STRL

Representations

The creative process maps can be represented graphically as well as be expressed algebraically.



The following algebraic expressions correctly describes the same maps:

$$P_I \widehat{=} \rightarrow_I ; (\rightarrow_E + \rightarrow_C)$$

$$P_{Ins} \widehat{=} \rightarrow_{Ins} ; (\rightarrow_{De} + \rightarrow_A + \rightarrow_D).$$

The map

$$P_{Ins} \widehat{=} \rightarrow_{Ins}; (\rightarrow_{De} + \rightarrow_{A} + \rightarrow_{D}).$$

was originally created from the *composition* of two sub-maps:

$$P_{Ins1} \widehat{=} \rightarrow_{Ins}; \rightarrow_{De}$$

and

$$P_{Ins2} \widehat{=} \rightarrow_{Ins}; (\rightarrow_{A} + \rightarrow_{D}).$$

using the distribution role of ; over the + which results in

$$P_{Ins} \widehat{=} P_{Ins1} + P_{Ins2}$$

We can also use this rule for *decomposing* a given map:

$$P_I \;\widehat{=}\; \rightarrow_I ; (\rightarrow_E + \rightarrow_C) \;=\; (\rightarrow_I ; \rightarrow_E) + (\rightarrow_I ; \rightarrow_C)$$

Creative Behaviours and Paths

- The traditional view of a behaviour is one of a sequence of states where the behaviours $< \sigma_1\ \sigma_2\ \sigma_3 >$, $< \sigma_1\ \sigma_3\ \sigma_2 >$ and $< \sigma_1\ \sigma_1\ \sigma_2 >$ are all distinct.
- Due to the multi-views nature of creativity, a single sequence of states will not adequately describe creative behaviours. Instead, a behaviour in our settings is treated as a **set of sequences of states,** where its elements reflect the various viewpoints of interest.
- A behaviour B is thus defined as $B \subset (\Sigma \times \Sigma)^*$.

STRL

Creative Behaviours and Paths - Cont.

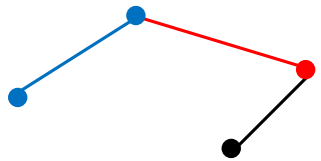
- $\{ < \textit{Contemplate Create Contemplate Acquiring} >, < \textit{Contemplate Create Acquiring Contemplate} > \}$ is a behaviour that is equivalent to
  $\{ < \textit{Contemplate Create Acquiring Contemplate} >, < \textit{Contemplate Create Contemplate Acquiring} > \}$

- Obviously,
  $\{ < \textit{Contemplate Create Contemplate Acquiring} >, < \textit{Contemplate Create Contemplate Acquiring} > \}$
  and
  $\{ < \textit{Contemplate Create Acquiring Contemplate} > \}$are two identical behaviours.

STRL

Creative Behaviours and Paths - Cont.

It is obviously clear that a creative path is a single behaviour.



Note here that a behaviour contains all or some of the viewpoints of interest. The above behaviour for example contains a recording of red, blue and black viewpoints. Additionally, we note that $(\Sigma \times \Sigma)^*$ gives the set of all possible behaviours or creative paths of the system.

STRL

Creative structures and Their Construction

- Let $V = \bigcup_i V_i$ be the set of all possible viewpoints of interest.

  We define a creative structure, $< C, \rightarrow_c >$, as

$$< C, \rightarrow_C > \widehat{=} < T, \rightarrow_T > \times \bigcup_i < V_i, \rightarrow_{V_i} >$$

  where

  - $< A, \rightarrow_A > \times < B, \rightarrow_B > \widehat{=} < A \times B, \rightarrow_A \times \rightarrow_B >$, and
  - $< A, \rightarrow_A > \cup < B, \rightarrow_B > \widehat{=} < A \cup B, \rightarrow_A \cup \rightarrow_B >$ such that $\rightarrow_A \times \rightarrow_B \widehat{=} \{< < a_1, b_1 > , < a_2, b_2 > > : \ a_i \in A, b_i \in B \ \wedge \ a_1 \rightarrow_A a_2 \ \wedge \ b_1 \rightarrow_B b_2\}$

Creative structures and Their Construction - Example

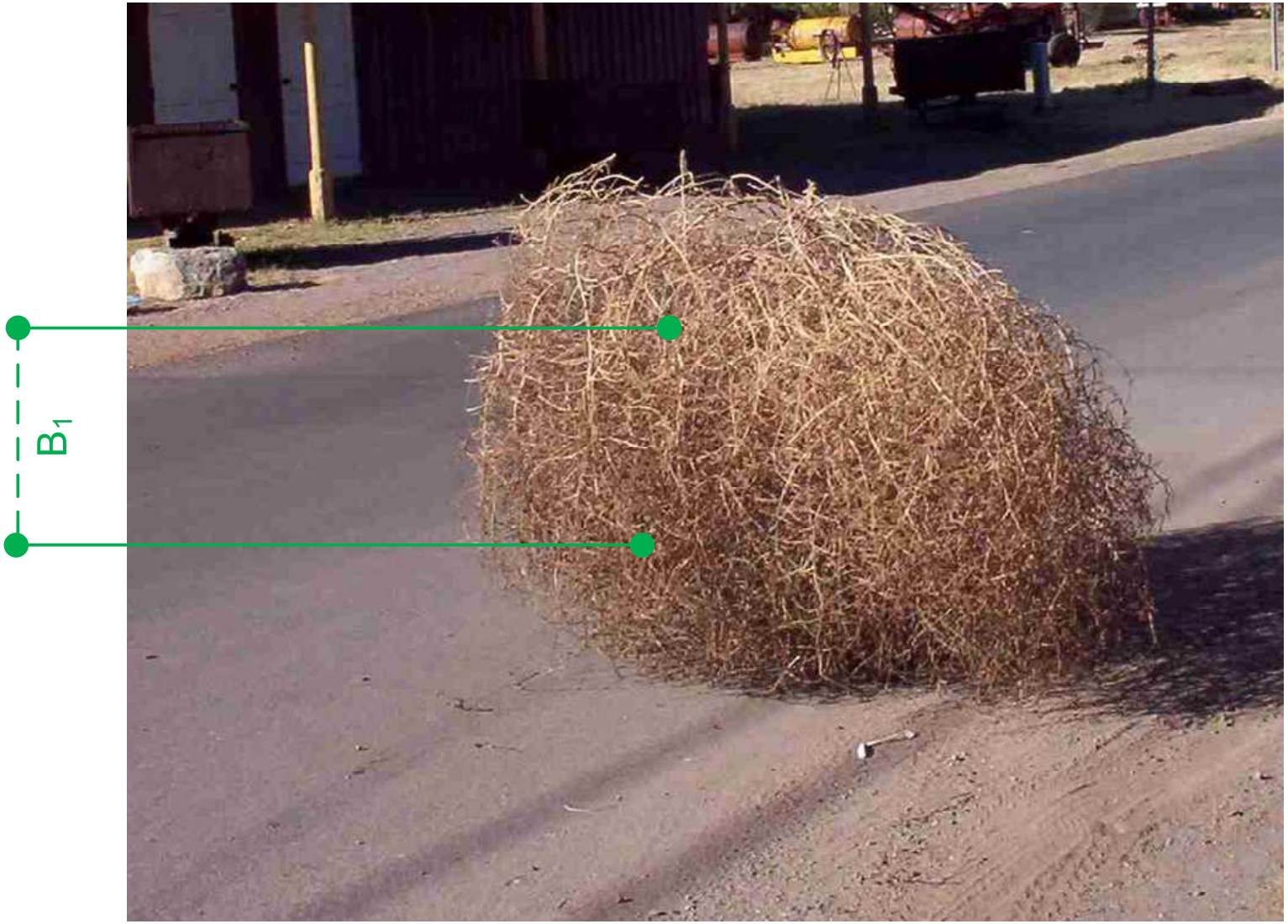
Let us consider three specific viewpoints of interest:

- Time - denotes the elapse time of the creation, with $< T, \rightarrow_T >$ as its transition system,
- Knowledge, represents the fact that the creator has gained knowledge (either by hopping from zone to another, contemplating, etc.), with $< K, \rightarrow_K >$ and
- Artifact- represents the process of creating or the production of the particular artifact with its system $< A, \rightarrow_A >$.

The resulting creative structure is thus defined and can be formed as:

$$< C, \rightarrow_C > \widehat{=} < T, \rightarrow_T > \times [< A, \rightarrow_A > \cup < K, \rightarrow_K >]$$

STRL



STRL

Operations Over Creative Structures and Co-creation

- The Operations over transitions presented above can be lifted over creative systems.
- Let $\Omega_1 \widehat{=} < A, \rightarrow_A >$ and $\Omega_2 \widehat{=} < B, \rightarrow_B >$ be two creative systems. These systems may represent either two different creations or a collaborative creation made by two different creators.
- $$\Omega_1 \ ; \ \Omega_2 \quad \widehat{=} \quad (\ \epsilon + \rightarrow_A ; (\rightarrow_A)^*) \quad ; \quad (\epsilon + \rightarrow_B ; (\rightarrow_B)^*)$$
$$= \quad \epsilon \quad + \quad (\rightarrow_A ; (\rightarrow_A)^*) \quad + \quad (\rightarrow_B ; (\rightarrow_B)^*) \quad + \quad (\rightarrow_A ; (\rightarrow_A)^*) \quad ; \quad (\rightarrow_B ; (\rightarrow_B)^*).$$

  One can observe that the composition could either be *empty* (ϵ), A, B or composition of two paths. If the composition is either A or B, hopping did not occur.

Operations Over Creative Structures and Co-creation - Cont.

- For a trueparallelism we have all possible combination from the two transition systems:
  $\Omega_1 \parallel \Omega_2 \quad \widehat{=} \quad < A \times B, \rightarrow_A \times \rightarrow_B >.$

- For interleaving semantics, we can take a transition from either A, B or both $((\rightarrow_A \wedge \rightarrow_B))$:
  $\Omega_1 \quad I \quad \Omega_2 \quad \widehat{=} \quad < A \times B, \rightarrow_A I \rightarrow_B >$, where
  $<\rightarrow_A I \rightarrow_B> \quad \widehat{=} \quad (\epsilon_A \wedge \rightarrow_A) \vee (\epsilon_B \wedge \rightarrow_B) \vee (\rightarrow_A \wedge \rightarrow_B)$

Creativity Laws - II

- $\Omega_1 / \Omega_2 = \Omega_2 / \Omega_1$
- $\Omega_1 \| \Omega_2 = \Omega_2 \| \Omega_1$

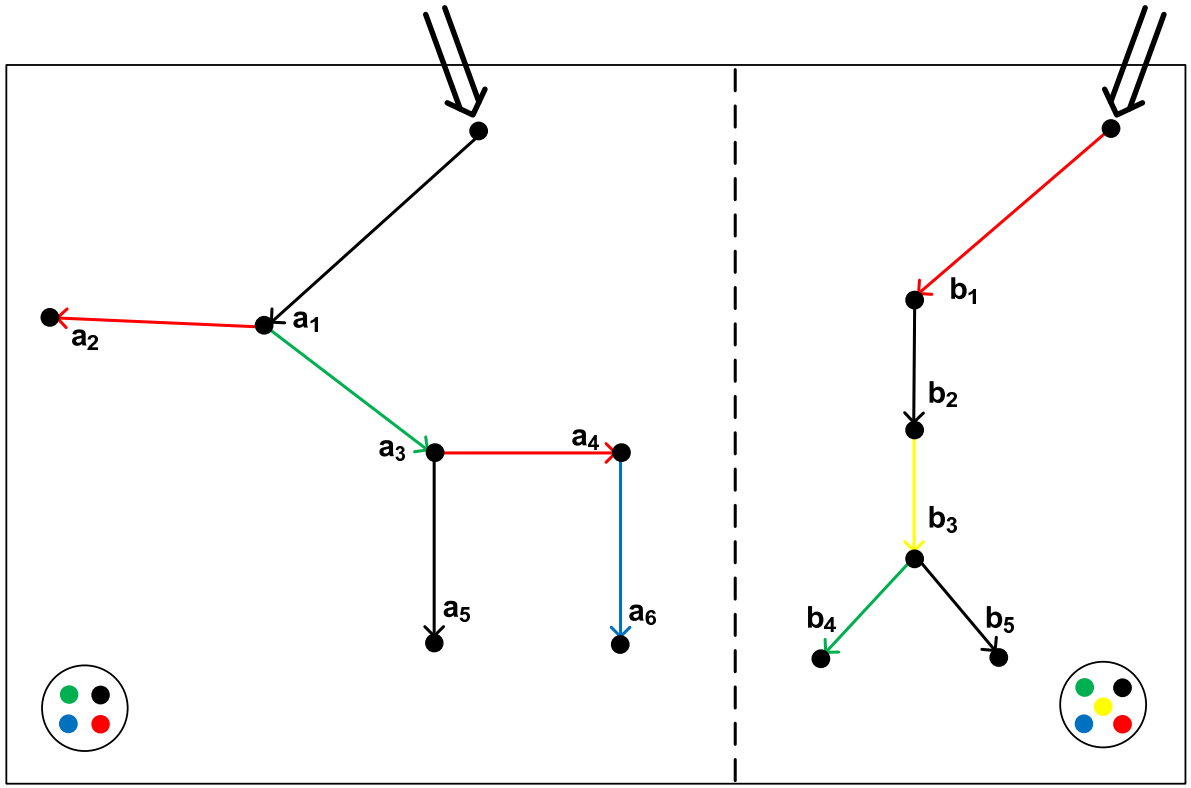
- $\Omega_1 / (\Omega_2 / \Omega_3) = (\Omega_1 / \Omega_2) / \Omega_3$
- $\Omega_1 \| (\Omega_2 \| \Omega_3) = (\Omega_1 \| \Omega_2) \| \Omega_3$

- $\Omega_1 ; (\Omega_2 / \Omega_3) = (\Omega_1 ; \Omega_2) / (\Omega_1 ; \Omega_3)$
- $\Omega_1 ; (\Omega_2 \| \Omega_3) = (\Omega_1 ; \Omega_2) \| (\Omega_1 ; \Omega_3)$

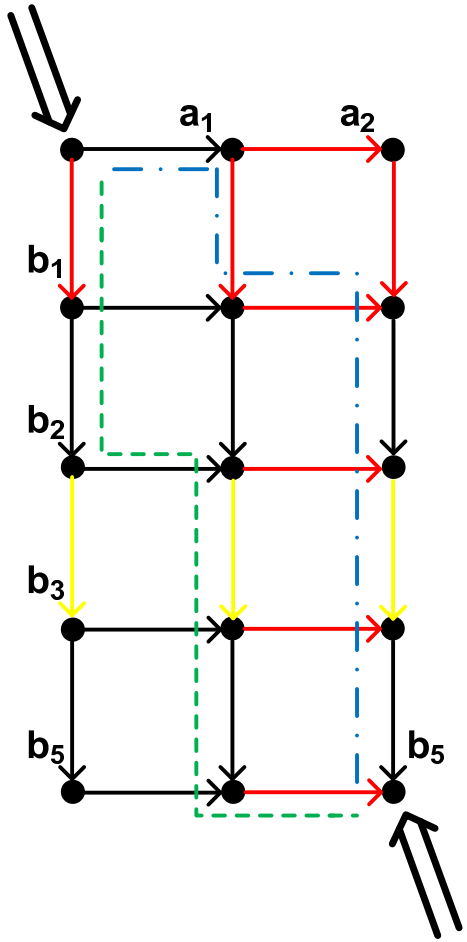
- $\Omega_1 ; (\Omega_2 ; \Omega_3) = (\Omega_1 ; \Omega_2) ; \Omega_3$

STRL

Collaboration



Analysis



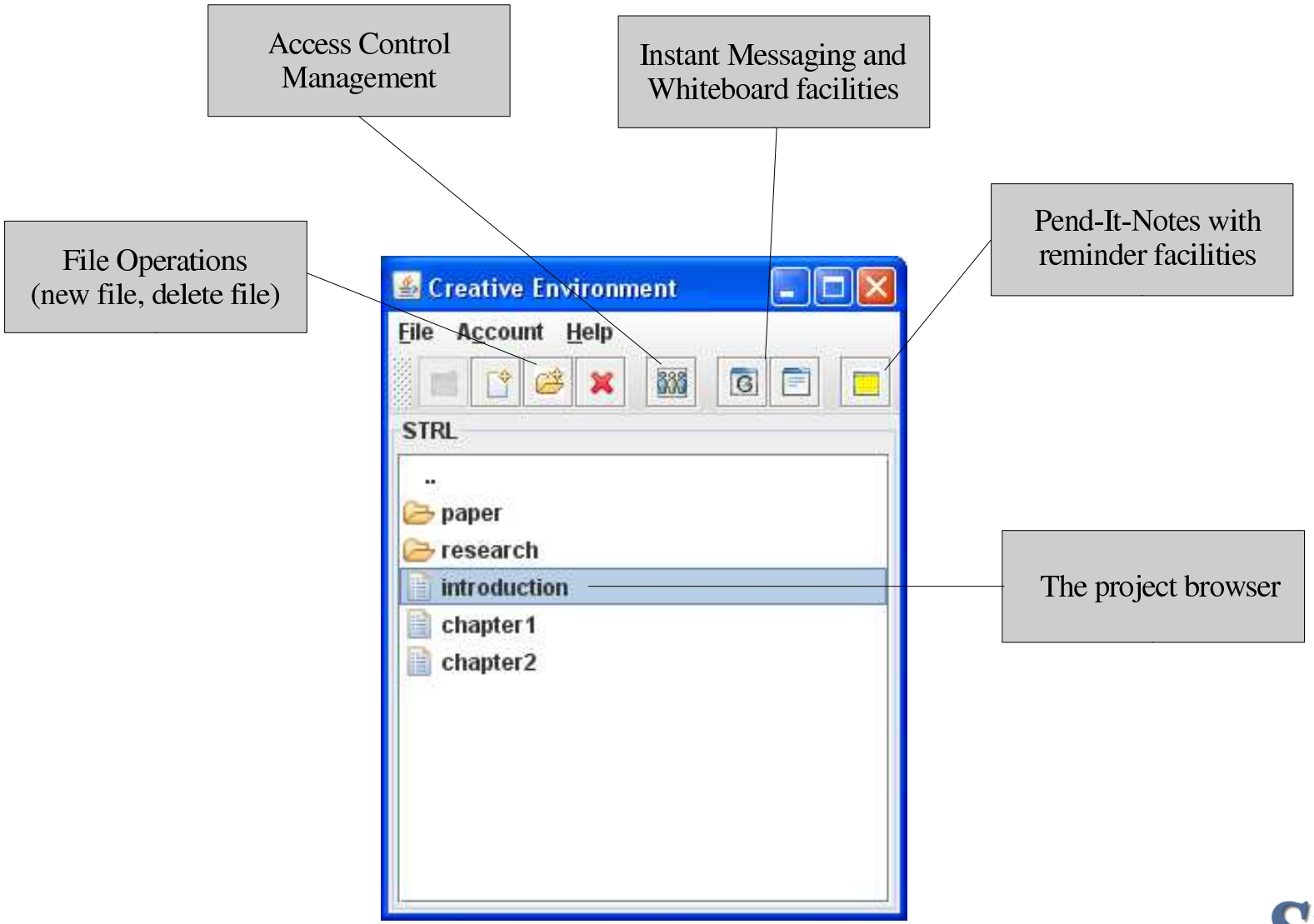
The De Montfort Creativity Assistant

The De Montfort Creativity Assistant is a tool set which was built to support the analysis of creative behaviours and processes. It has two major components:

- **De Montfort Creative Environment** and
- **De Montfort Creativity Mapper**

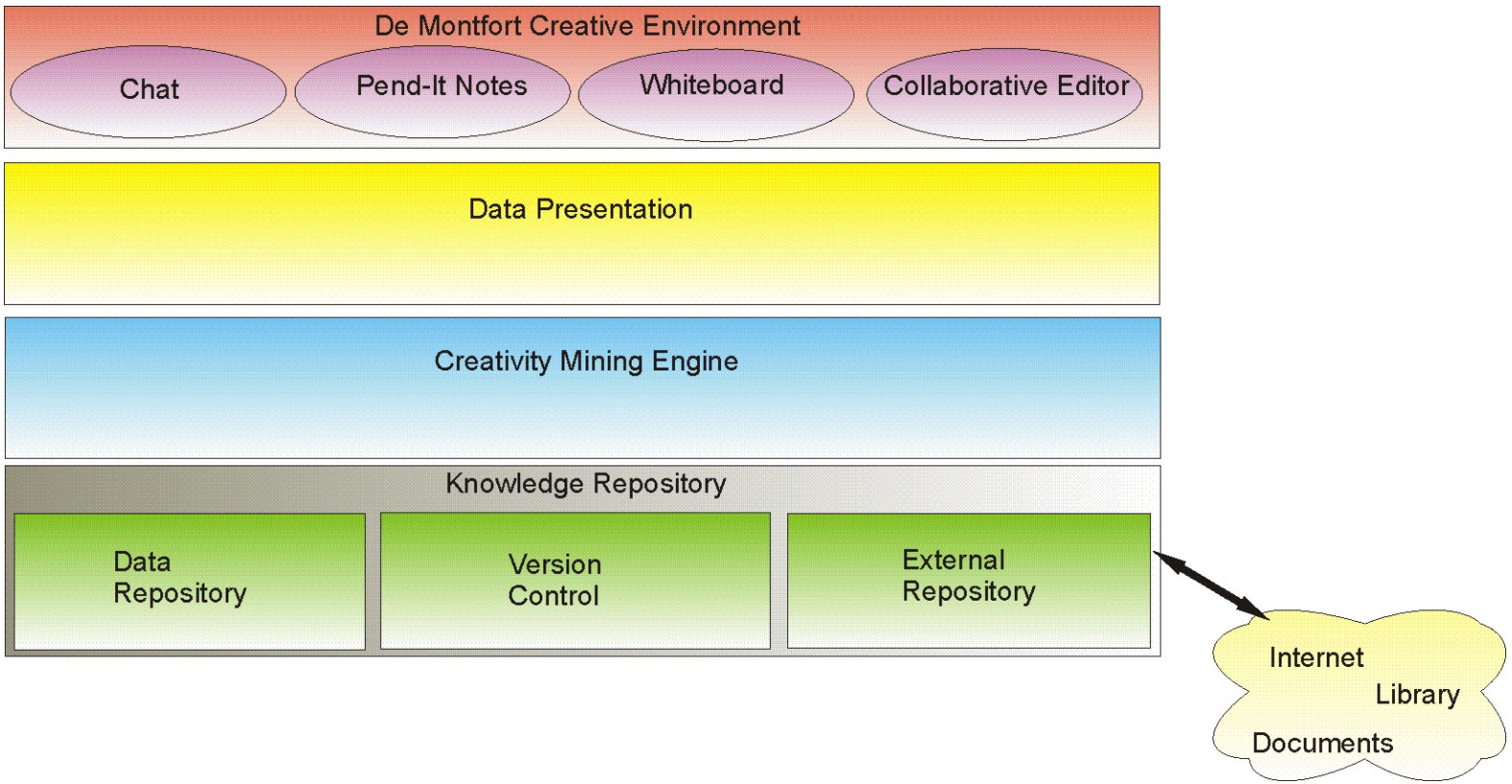
STRL

The De Montfort Creativity Assistant - Cont.



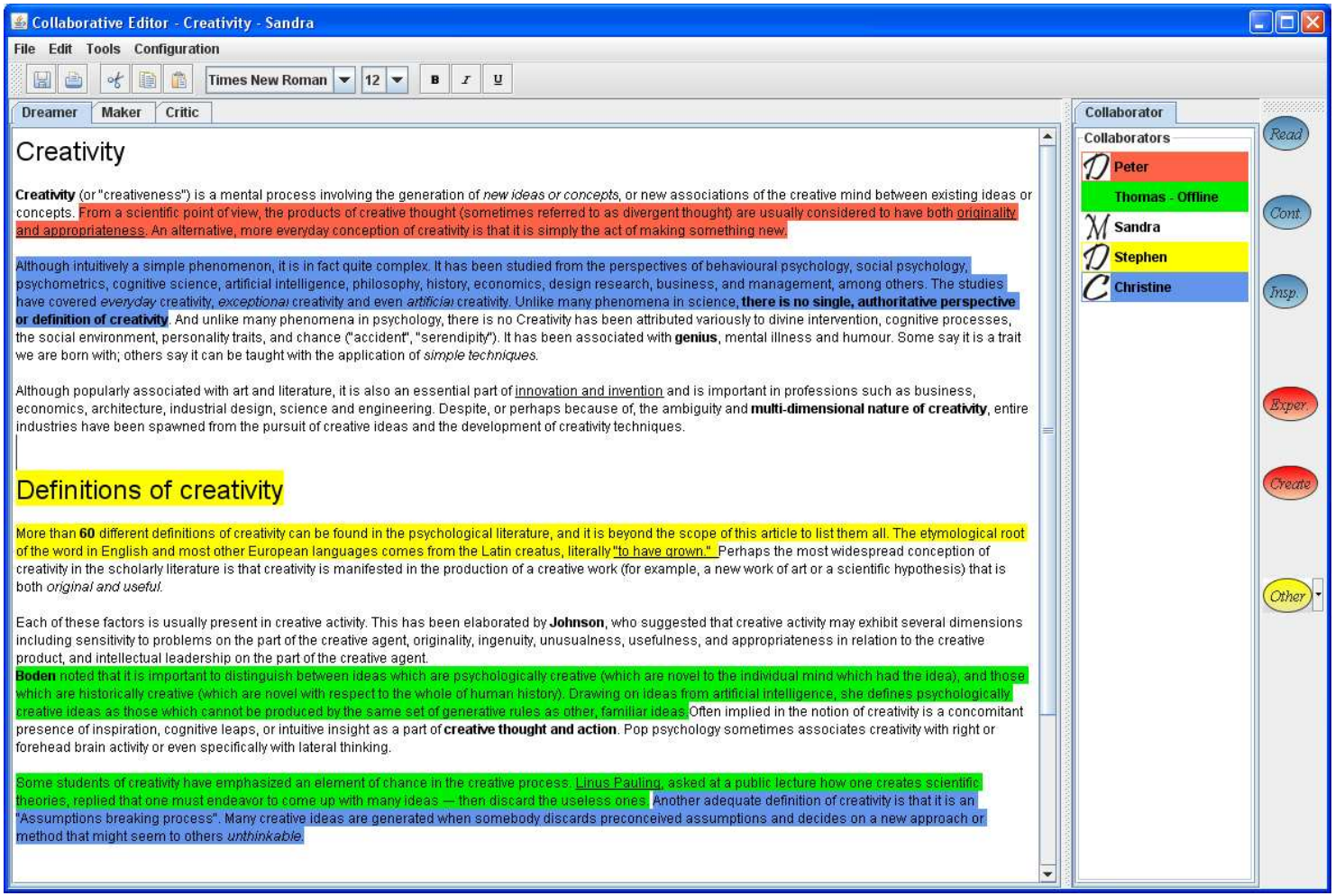
STRL

The De Montfort Creativity Assistant - Cont.



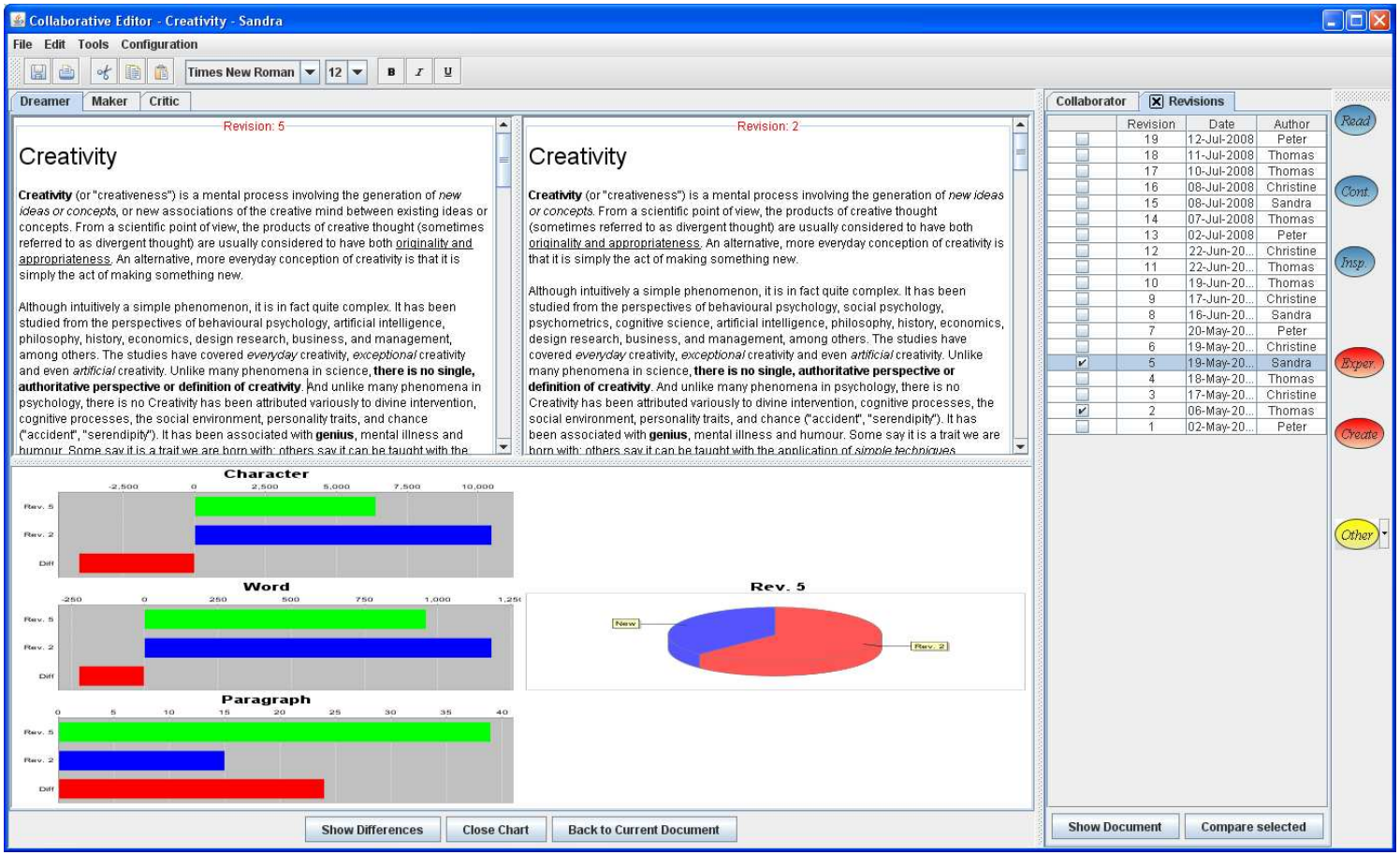
STRL

The De Montfort Creativity Assistant - Cont.



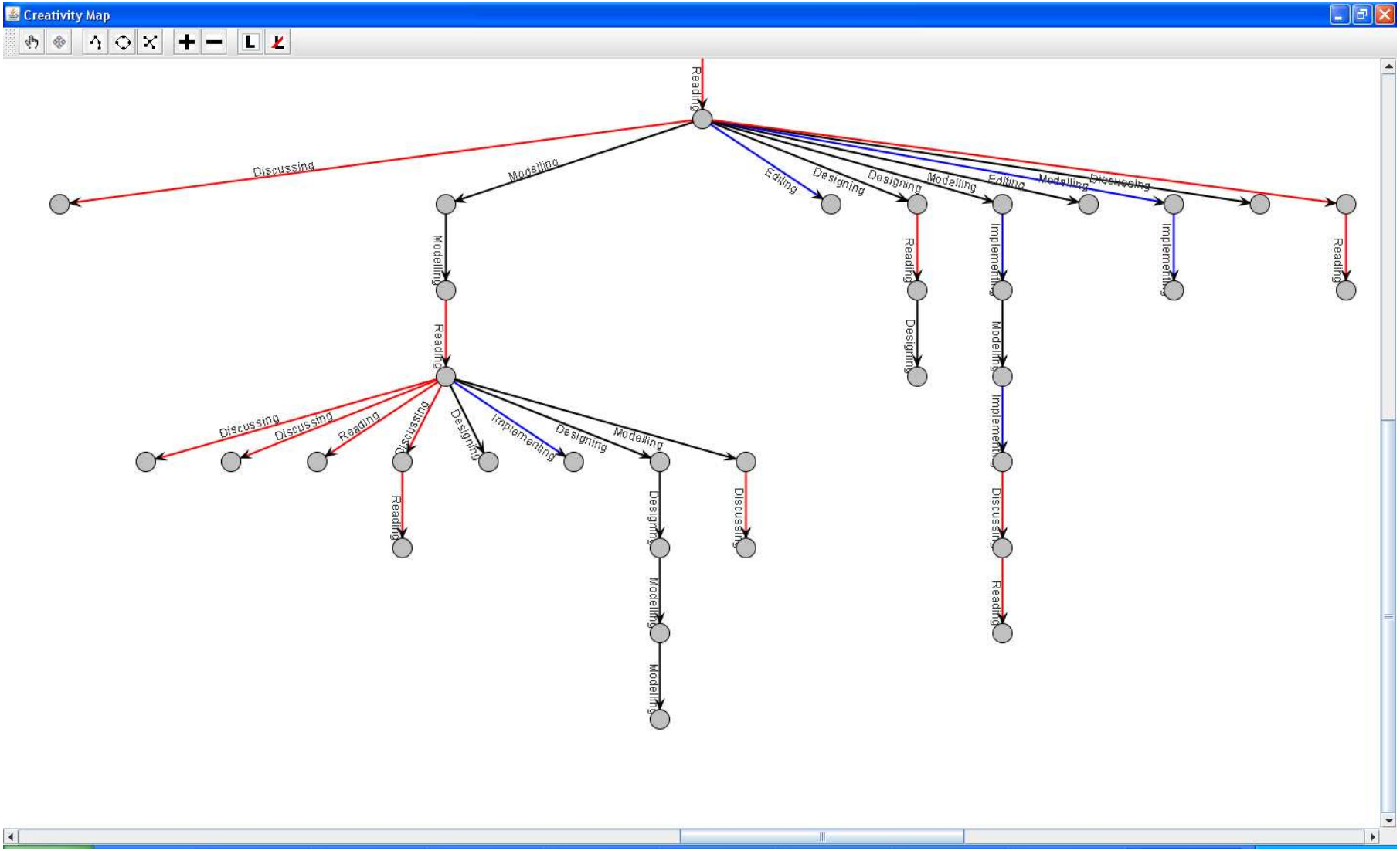
STRL

The De Montfort Creativity Assistant - Cont.

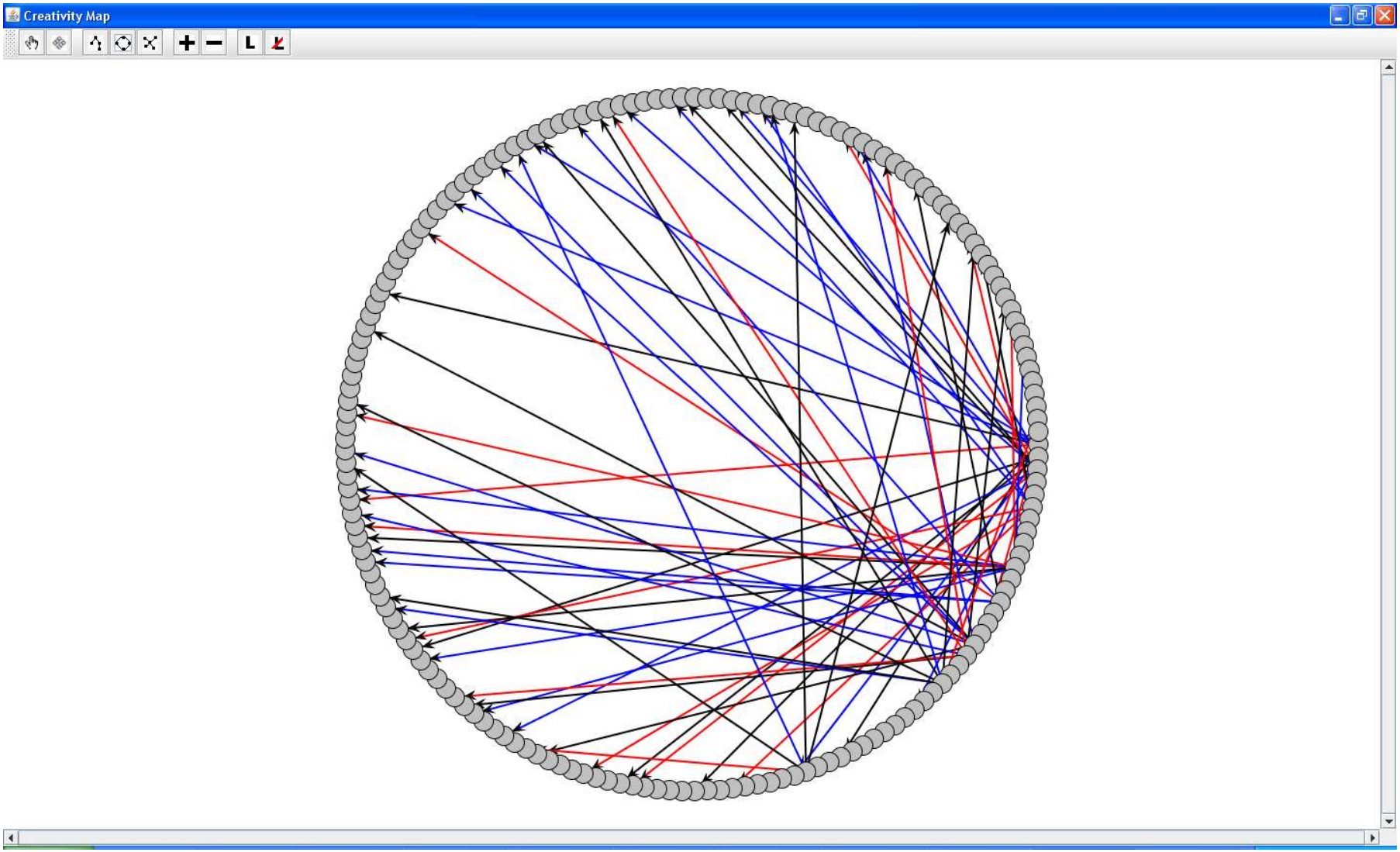


The De Montfort Creativity Assistant - Cont.

STRL

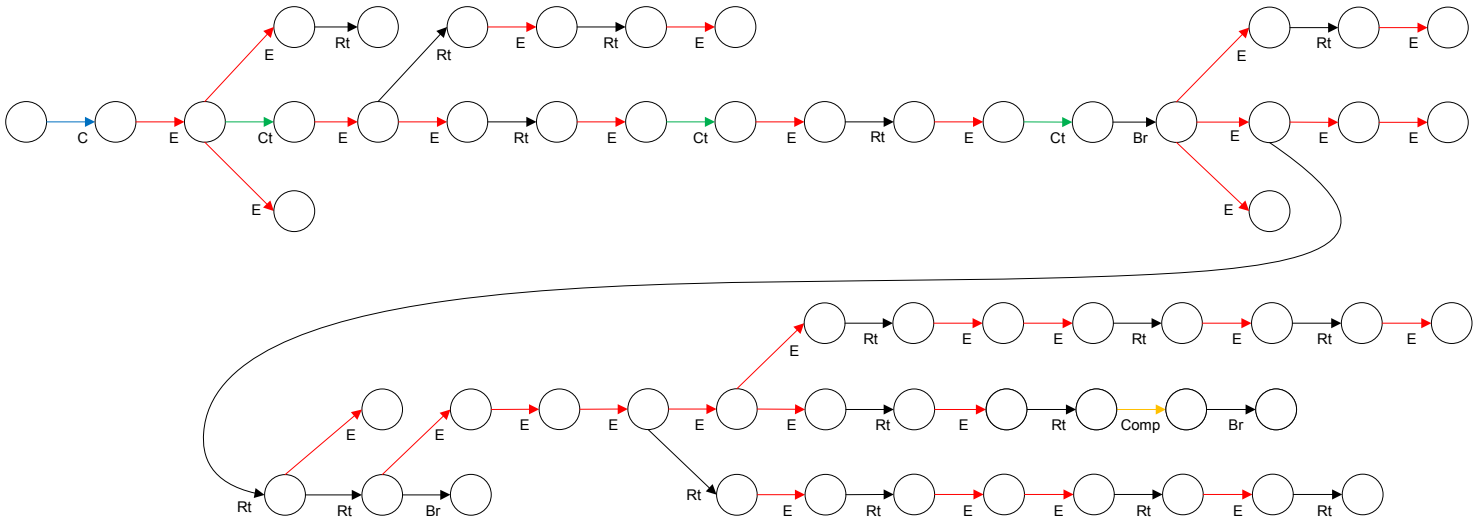

Creativity Map
Reading
Discussing
Modelling
Editing
Designing
Designing
Modelling
Editing
Modelling
Discussing
Modelling
Reading
Designing
Implementing
Modelling
Implementing
Discussing
Reading
Implementing
Reading
Discussing
Discussing
Reading
Discussing
Designing
Implementing
Designing
Modelling
Reading
Designing
Modelling
Modelling
Discussing

STRL



STRL

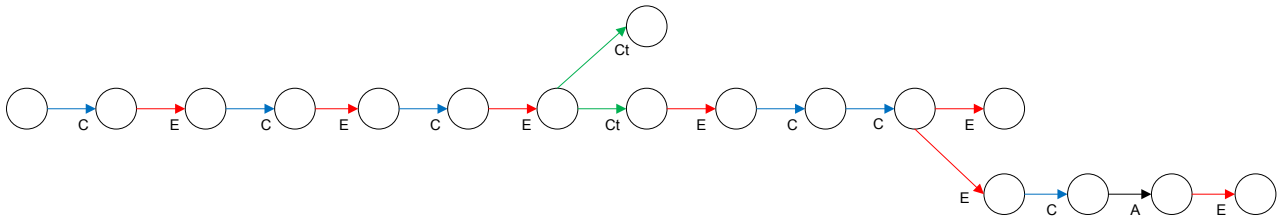
Creative Writing



Action	Description
E	Editing
Ct	Contemplating
C	Conceptualising
Comp	Comparing
Rt	Reading through
Br	Break

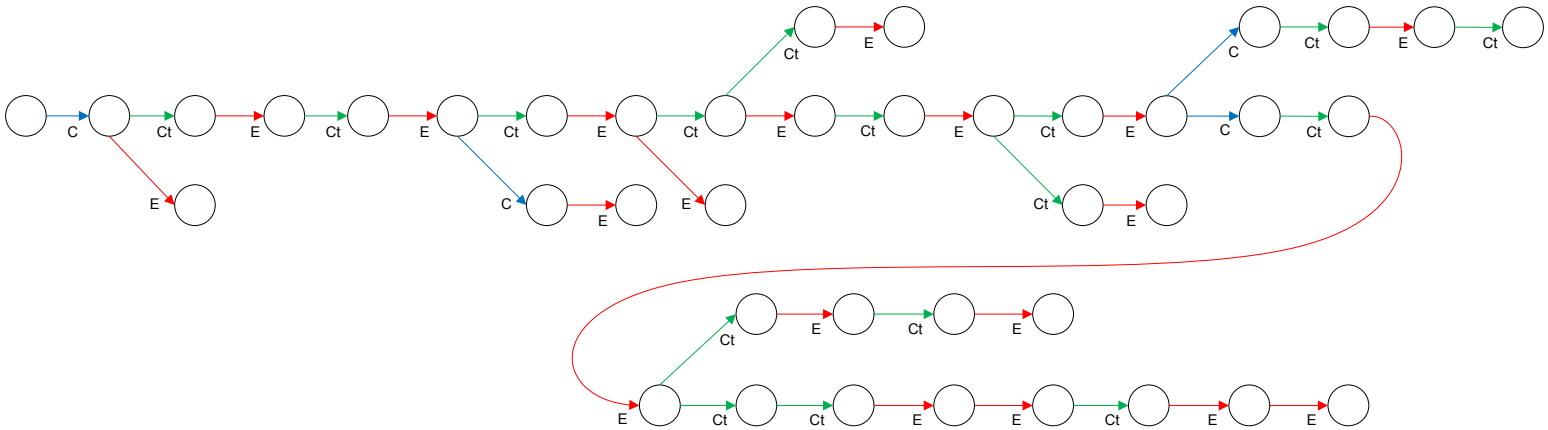
STRL

Creative Writing



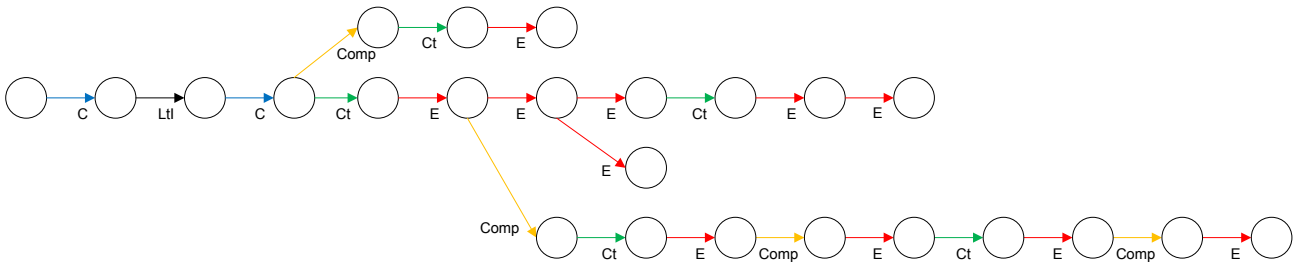
Action	Description
E	Editing
Ct	Contemplating
C	Conceptualising
A	Awaiting

Creative Writing



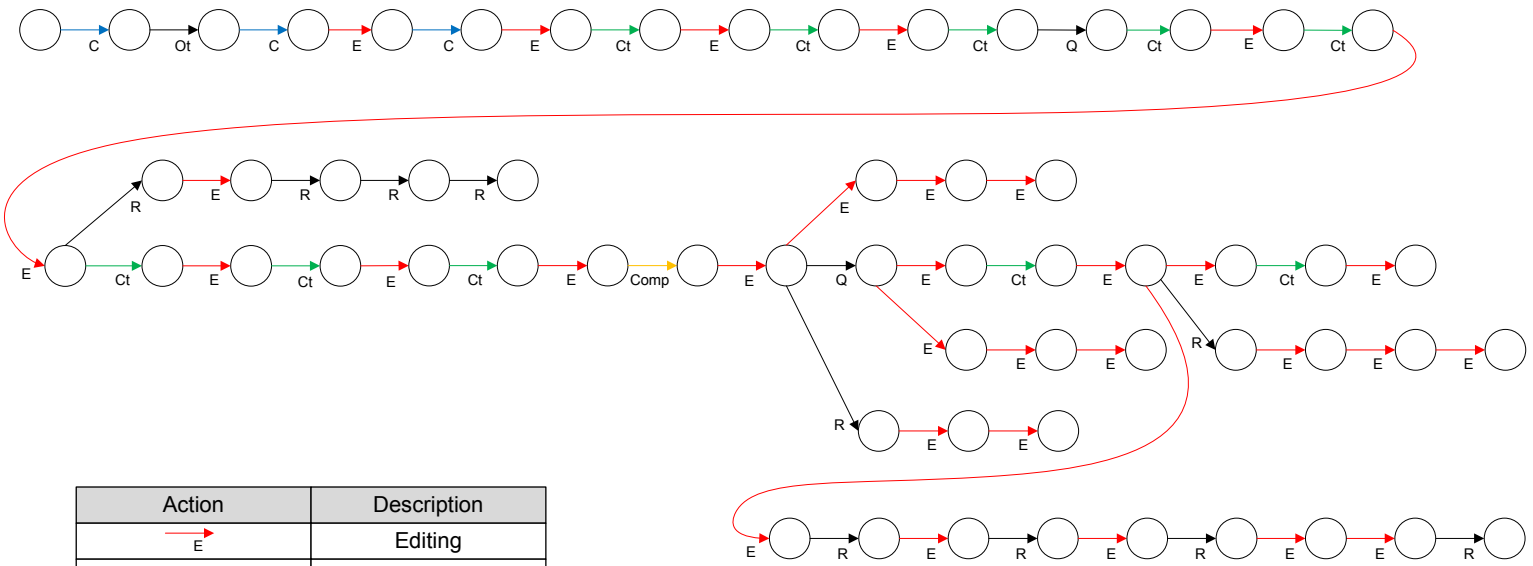
Action	Description
E	Editing
Ct	Contemplating
C	Conceptualising

Creative Writing



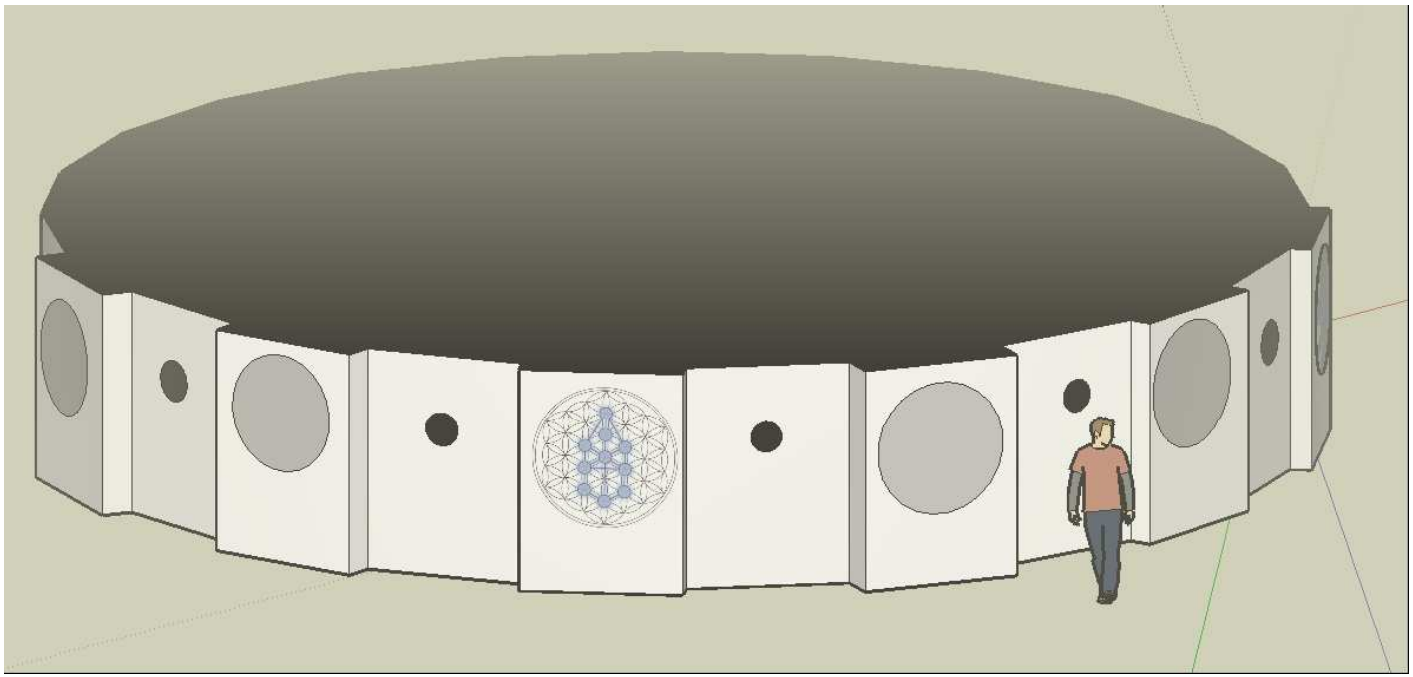
Action	Description
E	Editing
Ct	Contemplating
C	Conceptualising
Comp	Comparing
Ltl	Listening to Instructions

Creative Writing



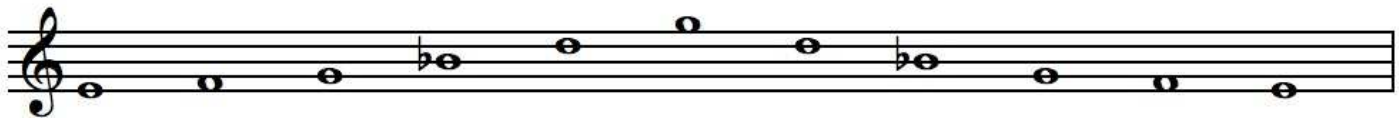
Action	Description
E	Editing
Ct	Contemplating
C	Conceptualising
Comp	Comparing
R	Reading
Q	Questioning
Ot	Other

Experiments: Music Composition



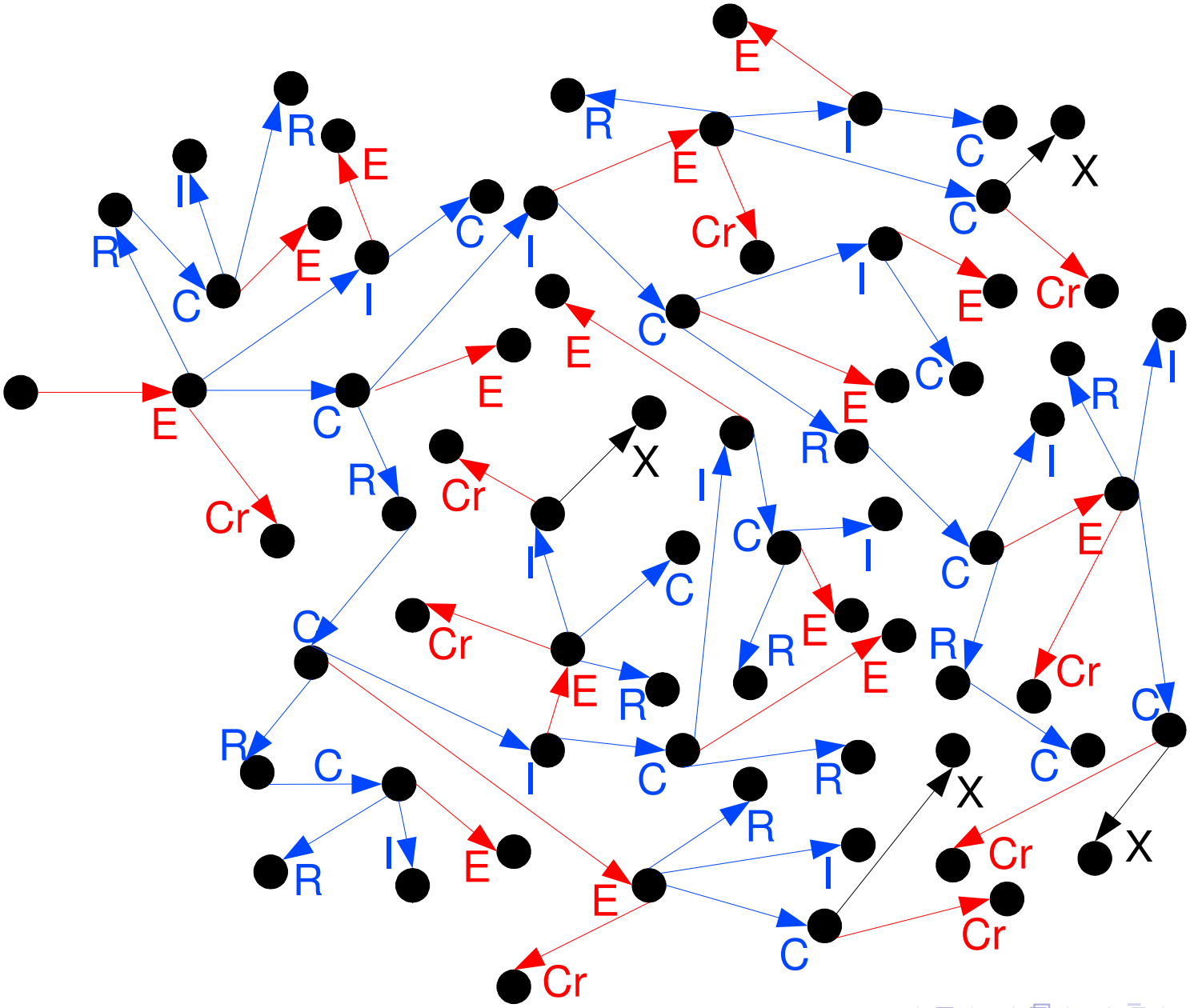
STRL

Experiments: Music Composition - Cont.



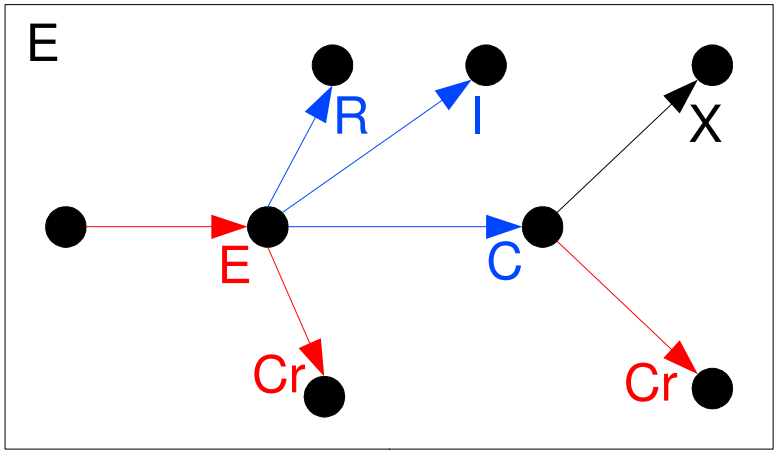
STRL

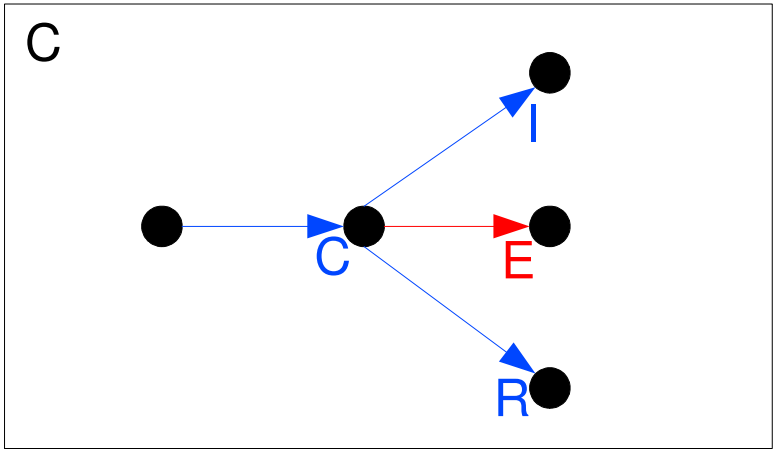
Experiments: Music Composition - Cont.

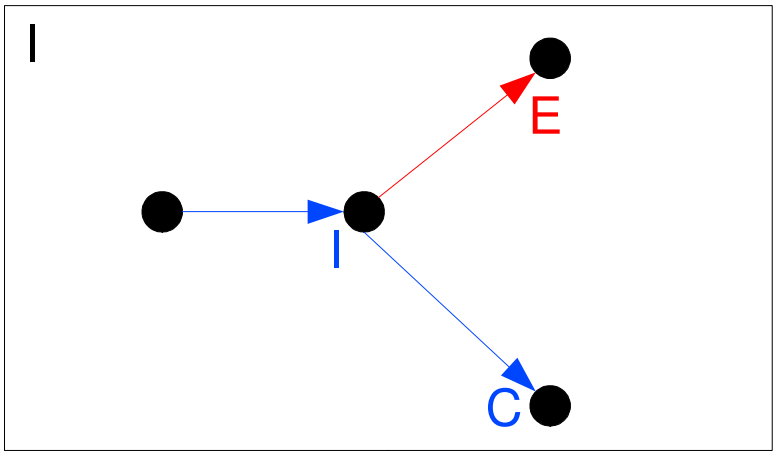


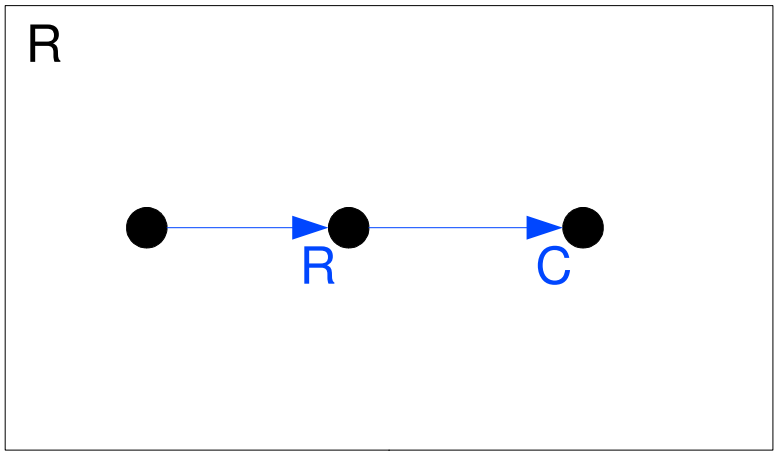
STRL

Experiments: Music Composition - Cont.



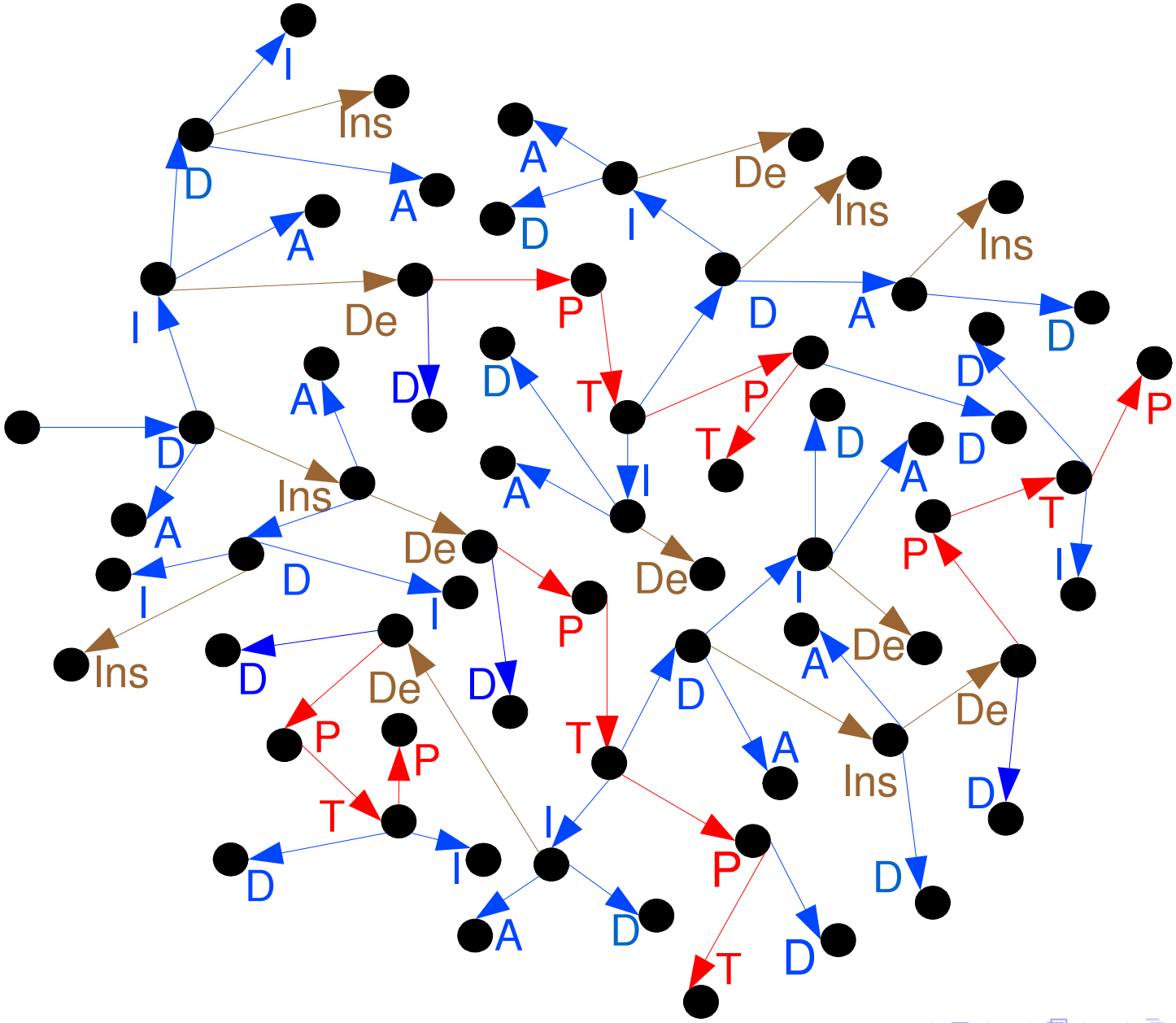






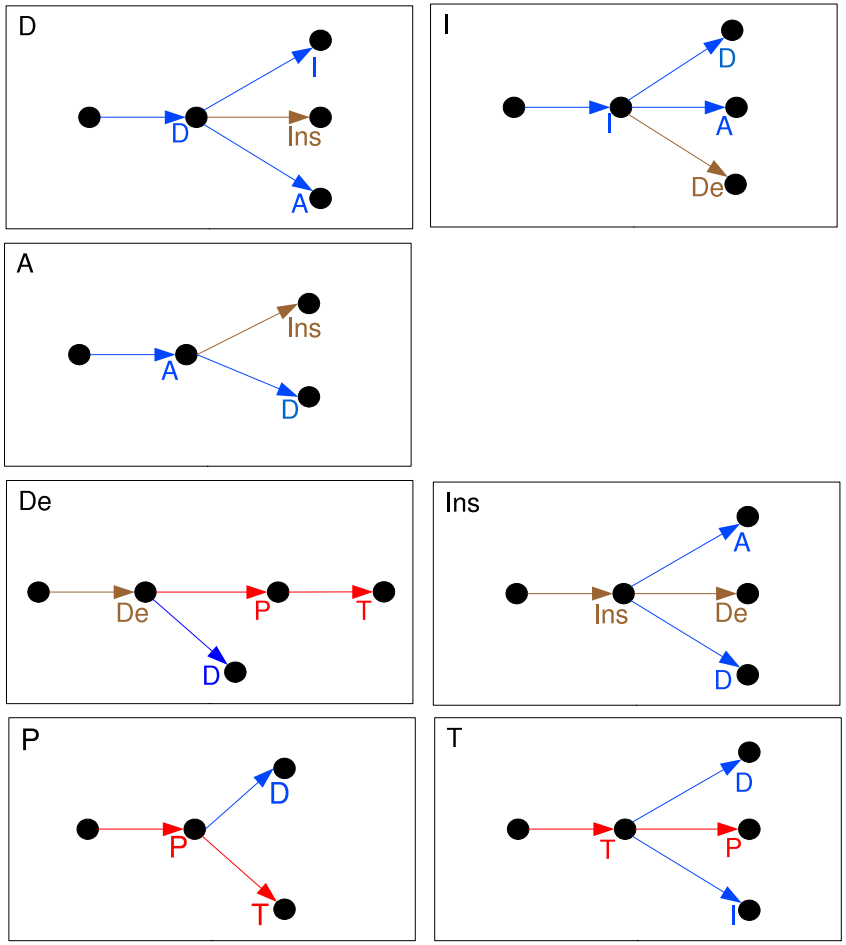
$$C_M = [T \parallel [R\ ;\ (C\ I\ I\ I\ E)]]^*$$

Experiment: Software Design - FermaT



STRL

Experiment: Software Design - FermaT



The whole $FeramT_C$ creative process can also be described as an expression in our algebraic setting as

$$FermaT_C = [T \parallel [N\ ;\ (IN\ /\ C\ /\ N)]]^*$$

Discussion and Insights

- Creativity Data Bank
- Creativity, Emergence and Diversity
- data, knowledge and information.

Discussion and Insights - Cont.

- Assigning 'importance' to creativity boxes allows us to study the frequency of its occurrence in that map and in other maps of the same and/or other creators within the same discipline.
- Similar findings may be obtained across different disciplines. This may reveal important knowledge about the creator, in addition to allowing us to build a taxonomy of creative processes.

STRL

Discussion and Insights - Cont.

- Once a statistically significant data set, a probabilistic creativity map for a given creator can be produced. Such a map will assign a probabilistic attribute to boxes and be able to infer some probabilistic properties about the creativity map.

- The choice of the labels, which describe the mood of the creator, presents us with the difficulties of **terminology alignment**. There are many existing techniques that can be use to address this issue utilising for example self organising map or alignment algorithms in the ontology domain.

STRL

Discussion and Insights - Cont.

- Each transition system in a creative map can be associated with an *action* set, which in turns adds more expressive power to our formalism.
  At a particular level of analysis, we may only be interested in discovering, for example
  1. the order of transitions, i.e. a *Create* activity is followed by three inspirational:
     $\rightarrow_{Create}$; $\rightarrow_{Inspiration}$; $\rightarrow_{Inspiration}$; $\rightarrow_{Inspiration}$
  2. the occurrence frequency of a particular transition (i.e. over the duration of the creation, 40% of transitions was $\rightarrow_{Prototyping}$).

- We can achieve this by decorating the transition by an appropriate action: $\rightarrow^{A}_{\alpha}$, e.g.

$$(\rightarrow^{Jogging}_{Inspiration} ; \rightarrow^{Praying}_{Inspiration}) ; \rightarrow_{Development} ; (\rightarrow^{Sleeping}_{Inspiration}) ; \rightarrow_{Development}$$

STRL

Educational Requirements

- These approaches introduce some important educational requirements of software engineering:
  1. **Provide a rich, more realistic and engaging development context.** For example, establishing hard deadlines, teaches the ability to frame problems and solutions more realistically while working in groups teaches working in teams.
  2. **Students should develop their own designs through rapid prototyping.** This will help identifying inconsistency and incompleteness in their ideas.
  3. **Use didactic method that supports creative thinking.**
- But
  1. what is **creative thinking**?
  2. How do we ensure it has been achieved?
  3. How do validate and verify its implementation?

STRL

Didactic Principles

- Support exploration. Let users try many alternatives before settling on a final design.
- Support low thresholds, high ceilings, and wide walls. Make it easy for beginners to start, but also let experts work on more complicated projects, and support a wide range of explorations.
- Support many paths and many styles. Assist learners with different styles and approaches.
- Support collaboration. Encourage teamwork.
- Support open interchange. Diverse tools that support creative work should be inter-operable.

STRL

Didactic Principles - Cont.

- Make it as simple as possibleand maybe even simpler. Avoid making tools too complex by adding unnecessary features.
- Choose black boxes carefully. Carefully select the primitives that users will manipulate.
- Invent things that you would want to use. Use your own experience in creative work.
- Balance user suggestions with observation and participatory processes. Involve end users in the design process.
- Iterate, iteratethen iterate again. Support iterative design using prototypes.
- Design for designers. Build tools that let others design.
- Evaluate your tools. Use empirical testing methods; do not rely on intuition.

STRL

- Software engineering educators can also use examples from interactive art projects and hacking to demonstrate innovation and nontraditional problem solving. To find these examples, we looked at electronic art conferences, such as Ars Electronica (www.aec.at), the Dutch Electronic Arts Festival (DEAF, www.deaf07.nl), and ACMs Multimedia Interactive Arts track. We also look at hacking conferences, such as Blackhat (www. blackhat.com) and Chaos Computer Congresses (www.ccc.de).