Roadmap to CRESY-II

ODD Protocol

Cara H. Kahl

Department of Social Psychology
FB Psych, Fak. EPB
University of Hamburg, Germany
cara.kahl@uni-hamburg.de

Contents

List of Figures ii

List of Tables iii

1 Roadmap to CRESY-II 1

1.1 ODD Protocol: Overview . . . 1
1.1.1 Purpose . . . 1
1.1.2 Entities, state variables, and scales . . . 2
1.1.3 Process overview and scheduling . . . 7
1.2 ODD Protocol: Design concepts . . . 7
1.2.1 Basic principles . . . 7
1.2.2 Emergence . . . 8
1.2.3 Adaptation . . . 8
1.2.4 Objectives . . . 9
1.2.5 Learning . . . 10
1.2.6 Prediction . . . 10
1.2.7 Sensing . . . 10
1.2.8 Interaction . . . 10
1.2.9 Stochasticity . . . 11
1.2.10 Collectives . . . 11
1.2.11 Observation . . . 12
1.3 ODD Protocol: Details . . . 12
1.3.1 Initialization . . . 12
1.3.2 Input data . . . 12
1.3.3 Submodels . . . 12
1.4 Model verification . . . 15
1.5 Independent variables . . . 17
1.6 Dependent variables . . . 17
1.6.1 hCAll . . . 17
1.6.2 hX.Y . . . 17
1.6.3 Creativity . . . 17
1.6.4 Reliability . . . 19
1.7 Experimental design . . . 19

Bibliography 20

Acknowledgements 24

List of Figures

1.1 CRESY-II: Sequence diagram 8
1.2 CRESY-II: Basic principles 9
1.3 CRESY-II: Pseudo code for submodel rate-patchwork 14
1.4 CRESY-II: Input parameters 18
1.5 CRESY-II: Experiments 19

List of Tables

1.1 CRESY-II Patchwork State Variables and Scales 3
1.2 CRESY-II Evaluator State Variables and Scales 4
1.3 CRESY-II Domain State Variables and Scales 5
1.4 CRESY-II Global Variables and scales 5
1.5 CRESY-II Dependent & Interim Variables and Scales 6

1

Roadmap to CRESY-II

CRESY-II stands for "CREativity from a SYstems perspective II", and it is the name of the second agent-based model conceived in this doctoral work. Similar to Chapter 1, the purpose of this chapter is to provide the reader with:

- A protocol of CRESY-II's architecture
- Information on the model's verification
- An overview of independent and dependent variables
- The design of experiments conducted with CRESY-II

The chapter starts with a detailed model protocol (Sections 1.1 - 1.3). The latter is designed according to contemporary guidelines for documenting simulation models ("ODD Protocol"; Grimm et al., 2006, 2010; Janssen et al., 2008; Polhill et al., 2008, 2010). It continues with information on the model verification process (Section 1.4). Following, CRESY-II's independent and dependent variables are explained (Sections 1.5 - 1.6). The chapter concludes with an overview of experiments conducted with CRESY-II (1.7).

1.1 ODD Protocol: Overview

Sections 1.1 - 1.3 contain a detailed account of CRESY-II's architecture. They follow the ODD (*Overview*, *Design concepts* and *Details*) protocol for describing individual-based and agent-based models (Grimm et al., 2006, 2010).

1.1.1 Purpose

The general purpose of this model, CRESY-II, is to simulate creativity as an emergent phenomenon resulting from creative production and evaluation processes exhibited by autonomous agents. The model demonstrates the effects, in terms of emerging product domains, of *creators* and *evaluators*. The former act on the basis of behavioural variability, i.e. a theoretical continuum describing how differently (variable) an individual behaves or creates from time to time (Stokes, 1999, 2007). The latter

act on the basis of varying levels of stringency regarding conventional evaluation criteria for creativity - novelty and appropriateness (Amabile & Mueller, 2007; Lubart, 1999; Metzger, 1986; Schuler & Görlich, 2006; Styhre, 2006; Preiser, 2006; Ward et al., 1999; Zysno & Bosse, 2009). An abstract model, CRESY-II was designed for theoretical exploration and hypotheses generation. It is the second model in a series designed to describe a systems approach to creativity in terms of variation, selection and retention subprocesses (Csikszentmihalyi, 1988, 1999; Ford & Kuenzi, 2007; Kahl, 2009; Rigney, 2001).

1.1.2 Entities, state variables, and scales

Programmed with NetLogo 4.1.2 (Wilensky, 1999), CRESY-II encompasses the following entities: creators, patchworks, evaluators, domain and global variables. The creators were carried over from CRESY-I. They are agents ("turtles" in NetLogo) characterized by the state variables described in CRESY-I's ODD Protocol. The only difference is that CRESY-II creators lack the variable *myPw*, which was originally programmed just for debugging purposes.

Patchworks have the basically same function as in CRESY-I. They abstractly represent artefacts creators produce and they are technically represented by stationary agents in NetLogo ("patches"). In CRESY-II, patchworks still appear as colours characterized by rgb values, so they are visible to the observer in NetLogo's View. Moreover, they are the means by which creators and evaluators communicate. Creators do not interact with their peers or evaluators directly. They gain knowledge of the domain by viewing displayed patchworks. The same holds for evaluators. As the addition of evaluators made new patchwork variables necessary, all state variables patchworks have are (re)listed in Table 1.1.

Table 1.1: *CRESY-II Patchwork State Variables and Scales*

Variable name	**Brief description**	**Value**
pcolor	List of rgb values (24 bit).	[r g b]
plabel	*domSize* category for *pcolor*	Integer, [0,*domSize*−1]
pDom	List of rgb values reduced to domain variable *domSize* (6, 9 or 12 bit).	[r g b]
pR, *pG*, *pB*	Patch's current respective red, green and blue values.	Integer, [0,255]
madeBy	Who made patchwork? Environment[1] or creator type.	{Env,Cx,C1,C2,C3}
whosNext	Holds *who* value of creator or evaluator potentially to next move on patch. Belongs to movement and evaluation procedures.	Integer, [1,number-creators][2] OR [1,number-evaluators][2]
justMade	Indicates whether patchwork was made during current step.	Boolean
toDo	Indicates whether evaluator still needs to judge patchwork.	Boolean
hue	Indicates whether patchwork's rgb colour is warm or cool.	{warm,cool}
currentEvals	List of evaluators current scores for given patchwork.	List with {0,1}
cScore	Patchwork's cumulative creativity score	Integer, [0,1][3]
PcScoreList	Current domain evaluation list of *cScore*s.	List of length *domSize* values of [0,1][3]

[1] "Environment" (Env) means a patch's colour has not been changed yet by any creator. It is still the colour it randomly received when the world was initiated.
[2] *number-creators* and *number-evaluators* are global variables which define the total number of creators (evaluators) in the world regardless of type.
[3] If a patchwork has never been rated before, its *cScore* is set to -1 by default.

Evaluators are mobile agents ("turtles" in NetLogo) characterized by the state variables described in Table 1.2, and their task is to judge the patchworks creators make. Note that in CRESY-II the variable *domSize* belongs these agents. Evaluators,

therefore, can only perceive patchworks with a certain degree of differentiation, and this recognition is not as fine-tuned as that of creators. In evolutionary terms, they only perceive a patchwork's phenotype, whereas creators are aware of patchworks' genotypes (Chattoe, 1998).

Table 1.2: *CRESY-II Evaluator State Variables and Scales*

Variable name	**Brief description**	**Value**
label	Describes (static) evaluator type.	{En,Ea,Ena}
eMem	Evaluator's memory list for patchworks seen. List of *domSize* nested lists. item 0 = *domSize value* (c_j), item 1 = *absolute frequency* (f_j) of value c_j.	[[0 f_1] [1 f_2],..., [c_j f_j],..., [n f_n]]
domSize[1]	Size of patchwork space, evaluators can perceive, e.g. number of rgb colours (patchworks) they can discriminate (12, 9 or 6 bit)	{64,512,4096}
novelty-stringency[1]	Percentage; How unknown may a patchwork be to be considered novel?	Integer, [0,1]
appropriateness-stringency[1]	Percentage; At least how appropriate should patchwork be to be considered so?	Integer, [0,1]
info-rate[2]	How many neighbours (4 or 8) an evaluator obtains info from per time step. Same for all.	{n4,n8}
movement[2]	How evaluators move in the world.[3]	{straightFd1, ahead3, allButBehind7, any8}

[1] This variable is set by the observer globally for all evaluators.
[2] In CRESY-II, the variables *info-rate* and *movement* are defined globally for all creators *and* evaluators.
[3] The movement strategies are explained in CRESY-I's ODD Protocol.

The domain is a higher-level entity abstractly representing a symbolic, or informational, system (Csikszentmihalyi, 1988, 1999). It is technically represented in NetLogo by a single, invisible agent (turtle), and its purpose is to record information about the patchworks. As some changes in code were made from CRESY-I to

CRESY-II, the domain consists of different variables in the later model. Table 1.3 gives an overview of them.

Table 1.3: *CRESY-II Domain State Variables and Scales*

Variable name	Brief description	Value
medallist	List of *domSize* nested lists containing all evaluations ever made for particular patchwork.	{0,1}[1]
scoreline	List of length *domSize* containing current creativity scores for patchworks.	Integer, [0,1][2]
just Rated	List of *number-creators* nested lists containing the creativity evaluations evaluators made in current step.	Integer, [0,1]
illustrated	List of length *number-creators* * 2 containing summed and squared creativity scores from *just Rated*.	≥ 0

[1] Note 0 = not creative, 1 = creative.
[2] Note a patchwork's *cScore* is set to -1 by default if it has never been evaluated before.

A number of global variables are used in CRESY-II to define start-up configurations (see CRESY-I's ODD Protocol for those carried over and Table 1.4 for new ones). Furthermore, CRESY-II's interim and output variables are defined as globals (see Table 1.5).

Table 1.4: *CRESY-II Global Variables and scales*[1]

Variable name	Brief description	Value
use-evaluators?	Observer decides this.	Boolean
number-evaluators	Observer defines number of evaluators in world	≥ 0
En, *Ea*, *Ena*	3 variables defining number of each evaluator type in world. Set by observer.	≥ 0
tickstop?	Observer decides if run should stop at *tickstop*	Boolean

[1] CRESY-II's globals consist of CRESY-I's *and* the ones listed here.

Table 1.5: *CRESY-II Dependent & Interim Variables and Scales*

Variable name	**Brief description**	**Value**
hX.Y	Mutual information with X = creator type & Y = patchwork type.	[0,1]
hX	Information measure (marginal entropy) for X = creator type.	≥ 0
hY	Information measure (marginal entropy) for Y = patchwork type.	≥ 0
hXY	Joint information (cell entropy)	≥ 0
hCx, hC1, hC2, hC3, hCAll	5 measures of relative information for each creator type and call creators	[0,1]
rel	Reliability measure.	[-1,1]
crea	Creators' overall creativity as judged by evaluators.	[0,1]
cond	Name of experimental condition.	String
rgbSpace	List of *domSize* nested lists. Each list contains rgb values according to *domSize* reduction.	Nested integer lists of length 3
rgbSpaceDom	List of *domSize* nested lists of length 2. Item 0 = *domSize* category; item 1 = 0 (default frequency).	Nested integer lists of length 2

CRESY-II is based on theoretical ideas expressed in creativity research in psychology. Therefore, there are no explicit concepts of spatial and temporal scales. However, research in this domain is usually conducted as psychological experiments in which individual participants or groups are requested to generate artefacts and independent raters are subsequently asked to judge them. Therefore, one time step in a simulation run can approximately equate to the time it takes to produce an artefact and to judge it. In CRESY-II, the duration of this time is the same for all creators and evaluators, which is not necessarily the case in reality. Furthermore, total duration (temporal extent) can also vary depending on whether psychological investigations take place in a laboratory setting, in which time is more or less controlled, or in the field where, for example, the non-simulated creative production and

evaluation processes carried out by artists, scientists, peers and the like can be of any duration and continuous, e.g. online ratings such as Amazon reviews.

In CRESY-II, the term *space* refers to knowledge and not to geographical location. Obviously, where creators and evaluators are located in the world will affect what knowledge (colour information) they acquire. However, as colours are randomly distributed on the grid when the world is initialized, geographical space has no systematic meaning in this model. By default, a 21*x*21 torus defines the knowledge-spatial extent of the world. Each patch on the torus represents one patchwork, i.e. a colour representing a creative product characterized by the three independent dimensions red, green and blue and the corresponding *domSize* category. The knowledge-spatial extent of the world is varied by altering the density of creators and evaluators in it.

1.1.3 Process overview and scheduling

CRESY-II consists of seven subsequent processes ("procedures" in NetLogo): obtain-info, make-patchwork, rate-patchwork, move, forget-some-info, tick, update-domain. Their scheduling is linear, i.e. their order occurs in the exact order they are listed in the sequence diagram in Figure 1.1. The agentsets running the commands in each process do so serially, but in a random order on the basis of NetLogo's *ask* command. As indicated in Figure 1.1, creators and evaluators obtain information, move and forget-some-information during the same stages of a simulation step. The state variables were updated asynchronously. Time was modelled discretely.

1.2 ODD Protocol: Design concepts

1.2.1 Basic principles

CRESY-II's overall architecture is based on a) the evolutionary mechanism of variation, selection and retention (VSR) as well as b) Csikszentmihalyi; Csikszentmihalyi's (1988; 1999) systems perspective of creativity which encompasses individuals, a field and a domain. CRESY-II's focus is on the interplay of variation and selection, including a simple form of retention. Various psychological models of the creative individual and their products (see also Ch. 4 in Sawyer, 2006) are used to develop the variation subprocess. The selection subprocess is modelled based on standard criteria used for evaluating creative products (Amabile & Mueller, 2007; Lubart, 1999; Metzger, 1986; Schuler & Görlich, 2006; Styhre, 2006; Preiser, 2006; Zysno & Bosse, 2009) and Amabile's (1996) Consensual Assessment Technique (CAT). The domain, a higher-order entity representing the process of retention, is characterized by a measure of information (qualitative or nominal diversity) borrowed from Shannon's (1948) mathematical theory of communication. The domain's implementation is additionally based on the notion that the more an artefact is valued, the longer it remains in the domain. Figure 1.2 summarises CRESY-II's basic principles graphically.

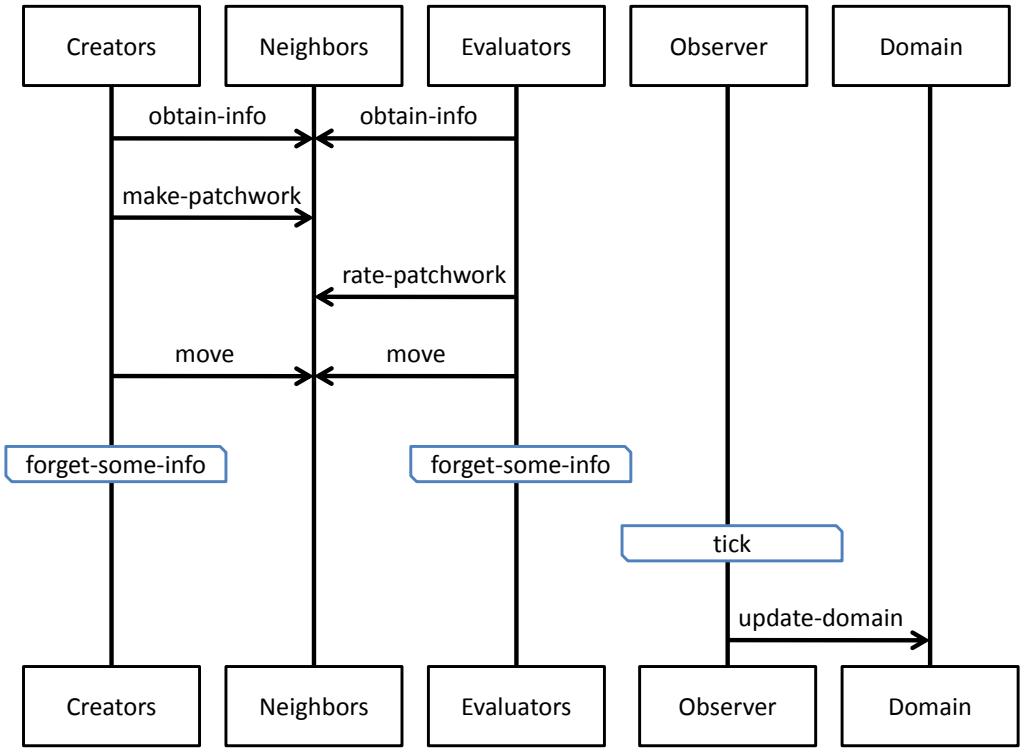

Figure 1.1: *Sequence diagram of CRESY-II.*

1.2.2 Emergence

Please review CRESY-I's ODD Protocol, as the notes on emergence in CRESY-I still hold in CRESY-II. Additionally, the two dependent variables *rel* and *crea* are modelled as emerging microsystem characteristics based on evaluators' behaviour. Both measures are expected to vary in magnitude and volatility depending on initial start-up conditions as well as activity exhibited by creators and evaluators. *rel* measures the reliability of evaluators' ratings. It is therefore an indicator of how similar these agents are in behaviour. Do they represent a homogeneous or heterogeneous microsystem? The measure allows the modeller to reflect their collective behaviour. *crea* measures how creative the evaluators find creators' patchworks in a given step. It therefore reflects the quality of artefacts, and in an aggregated form it gives way to diverse interpretations about the (sub)systems: If it is high, for instance, does it mean that creators are performing well or that evaluators are easy to please? Or could it mean the environment is just right for inducing high levels of creativity?

1.2.3 Adaptation

Please refer to CRESY-I's ODD Protocol to see how creators adapt. Additionally in CRESY-II, creators adapt to the constraints of the domain by having to paint their patchworks on neighbouring patches with the lowest creativity scores. In this way, they are subject to the most valued works for a longer time period. Evaluators adapt

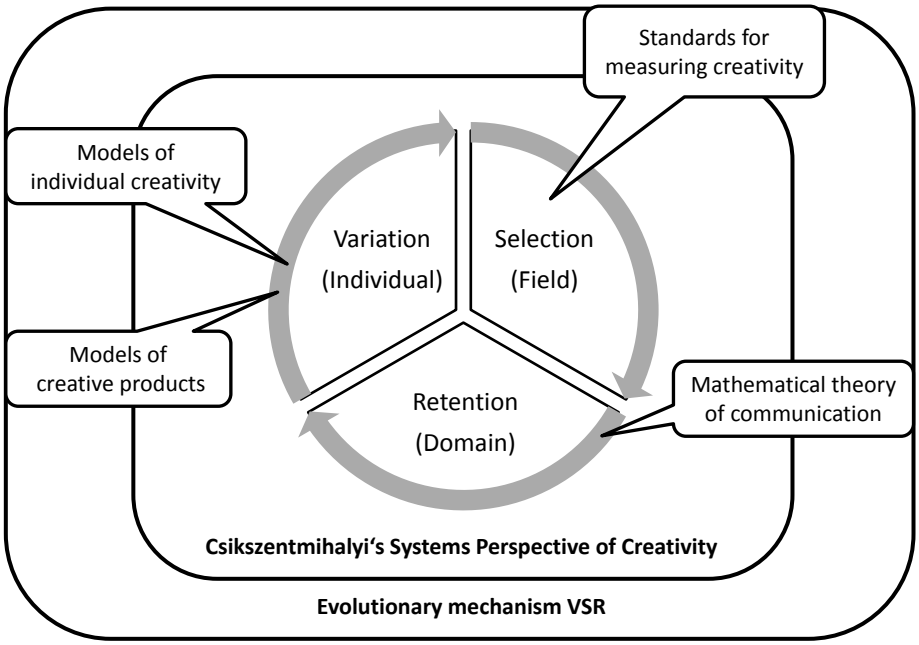

Figure 1.2: *Basic concepts and models used in CRESY-II: Evolutionary mechanism VSR, Csikszentmihalyi's (1999) systems perspective of creativity, psychological models of individual creativity, psychological models of creative products, psychological standards for evaluating creativity and the mathematical theory of communication proposed by Shannon (1948).*

to their environment by obtaining information every step, which in turn affects the way they judge patchworks' novelty. They further adapt when rating appropriateness by viewing a part of the context a specific patchwork is embedded in. Evaluators do not explicitly seek to increase their individual success, but their consistent behaviour based on their adapting memories and the changing contexts surrounding patchworks is expected to contribute to the degree of creativity (*crea*) a domain is assessed to have.

1.2.4 Objectives

Creators' objectives were stated in CRESY-I's ODD Protocol and were not changed in the current model. Evaluators have two implicit objectives: 1) They are supposed to judge the creativity of patchworks, and 2) they are to do so based on their individual evaluator type, therewith incorporating the criteria novelty and appropriateness in their ratings. They achieve the first objective by giving a patchwork a zero if considered not creative, else a 1. The second objective is actually a prerequisite for the first, and the behavioural rules are set by the observer. Evaluator type *En* only uses perceived novelty as a basis for creativity ratings, *Ea* only appropriateness and *Ena* both. How evaluators meet these objectives is measured collectively with

the dependent variables *crea* (creativity; objective 1) and *rel* (reliability; objective 2). Note that both creators and evaluators do not adapt their behavioural rules (or types) to the environment, although their knowledge base does change over time. On a whole, however, the former act to increase diversity and the latter to qualify it in the domain.

1.2.5 Learning

In CRESY-II, creators and evaluators learn by "memorizing" or obtaining patchwork information from neighbouring patches. Their memories, i.e. adaptive traits, are also affected by their random forgetting of what they have encountered before (see also submodels *obtain-information* and *forget-some-info* in CRESY-I's ODD Protocol). Both entities, however, do not learn to alter their behavioural rules.

1.2.6 Prediction

The concept of prediction does not appear to apply to CRESY-II.

1.2.7 Sensing

In CRESY-II, creators and evaluators only sense parts of the environment locally by viewing different patchworks based on their movement strategy and by saving patchwork information in their memory lists. While creators can produce and view all possible red, green and blue values (0-255 per colour; 24-bit total colour space), evaluators cannot discriminate this detail with which creators produce patchworks. They only perceive a reduced colour space according to *domSize* (see Table 1.2 and the submodels in Section 1.3.3). This is comparable to consumers, users, etc. viewing end products without recognizing all their technical details. It is also a way of implementing the notion that evaluators only perceive the phenotypes of products, while creators are aware of their genotypes. Sensing also applies to the way creators place patchworks they currently make. A simple way to formalize retention, creators place their newly made patchworks on a neighbouring patch with the lowest creativity score (*cScore*). This ensures that the most valued patchworks remain in the domain longer. By sensing these scores, creators are influenced by evaluators' ratings.

1.2.8 Interaction

Creators and evaluators experience indirect interaction with other creators by locally viewing and obtaining information about their behaviour, i.e. patchworks. This feature is comparable to situations in which creators do not know each other personally, but know each other's works (Dennis & Williams, 2003; Müller, 2009; Runco et al., 1994; van den Besselaar & Leydesdorff, 2009), and this situation is common in creativity research (Kozbelt & Serafin, 2009). It is also comparable to the way products are rated by judges in creativity research. Raters usually do not have contact with

study participants. They judge the products independently and anonymously (Bechtoldt et al., 2010; Kaufman et al., 2008; Lonergan et al., 2004; Rietzschel et al., 2006; Silvia, 2008).

1.2.9 Stochasticity

The areas mentioned in CRESY-I's ODD Protocol still hold for CRESY-II. In addition to them, stochasticity also applies to the following areas. *Dispersion of evaluator types*: Evaluators are always randomly dispersed on the simulation grid when the world is initiated. *Making patchworks III* (continued from CRESY-I's ODD Protocol): When a creator places a patchwork on a neighbouring patch, it first determines all patches with the same lowest *cScore*. It then randomly chooses one of them to place the new patchwork on. Note in some cases there could be only one neighbouring patch to choose from. *Rating patchworks*: Stochasticity applies to the way appropriateness is assessed. In doing so, an evaluator looks at the hues of six of eight neighbouring patches. The six neighbours are chosen randomly, meaning not all evaluators use the same context as the basis for their appropriateness judgements. *Forgetting information*: Similar to the way creators forget, three *domSize* categories with absolute frequencies larger than zero are independently and randomly chosen from an evaluator's memory *eMem*. Each category's absolute frequency is then reduced by 1.

1.2.10 Collectives

The collectives in CRESY-II do not refer to emergent properties of individual agents, but to the kinds of entity groups the model encompasses. As collectives, there are creators, evaluators, patchworks and a domain. Moreover, there are creator subtypes (*Cx*, *C1*, *C2*, *C3*) which differ in the way they make patchworks (see submodel *make-patchwork* in CRESY-I's ODD Protocol). The evaluators form another entity group, and three types are available (*En*, *Ea*, *Ena*). Creators and evaluators are therefore separate kinds of entities with their own state variables and traits. Although they do not communicate directly with each other, they are affected by each other's behaviour. The information creators and evaluators obtain during the simulation (procedure *obtain-info*) is, especially as a run prolongs and with a large number of creators, produced by the collective of creators. Moreover, creators place their patchworks on patches with the lowest creativity scores (*cScores*), meaning they are affected by evaluators' work. So although both collectives are not characterized by direct intra- and interaction, they are influenced on a whole by what every agent is doing. This leads to the conclusion that they are affected by each others' work, i.e. patchworks and evaluations. These two things are what constitute the domain, a dynamic second-order entity whose characteristics are ultimately expressed in the dependent variables *hCAll* and *crea*.

1.2.11 Observation

Data collection is conducted in two ways. Firstly, diverse monitors and plots are included in the NetLogo Interface Tab to allow graphical observation simultaneously while a simulation is running. There are plots for the following information: *hCAll*, *hX.Y*, *rel*, *crea*, domain creativity scores (*cScores*) as well as for the patchwork distributions of all creators and each creator type separately. There are monitors for the dependent variables *hCAll*, *hX.Y*, *rel*, *crea* and the interim variables *hX*, *hY* and *hXY* (see Section 1.6 and CRESY-I's ODD Protocol for more on these variables).

Secondly, numerical data collection was achieved by designing experiments with NetLogo's BehaviorSpace (Wilensky, 1999). Each experiment produced a *.csv* file containing values for all dependent variables (Table 1.5) at every time step of each simulation run. The data was used freely to analyse CRESY-II, although not all variables were selected for model analysis. All dependent variables are taken from the observer perspective.

1.3 ODD Protocol: Details

1.3.1 Initialization

All variables used for initialization were derived from theoretical deliberations or empirical findings discussed in creativity research (see for example Ochse, 1990; Sawyer, 2006; Sternberg, 1999; Sternberg et al., 2004; Zhou & Shalley, 2007). Their exact values were initially arbitrarily set and then explored in the experiments described in Section 1.7. Please refer to each experiment's documentation for the exact initial settings (see also Figure 1.5).

1.3.2 Input data

CRESY-II does not use input data to represent time-varying processes. The model is an abstraction derived from normative ideas published in creativity research chiefly within the field of psychology.

1.3.3 Submodels

This section describes CRESY-II's processes or *submodels* (see also Figure 1.1). If they function the in same way as in CRESY-I, they are not re-explained here but cross-referenced to CRESY-I's ODD protocol (Chapter 1).

obtain-info

Creators and evaluators obtain information in the same way as in CRESY-I (see the latter model's ODD Protocol). The only difference is that evaluators save patches' *plabel* variable, which indicates their *domSize* category. This does mean creators

have more (memory) variables than evaluators, and at face value they receive more information. However, each memory variable - be it that of creators or evaluators - is updated with the same amount of information according to *info-rate*[1]. Creators' state variables *cR*, *cG*, and *cB* (see CRESY-I's ODD Protocol) and evaluators' state variable *eMem* (Table 1.2) are updated in this procedure.

make-patchwork

Creators make patchworks in exactly the same manner as in CRESY-I (see Section **??**). The only difference is that creators no longer place their patchworks on the patches they are currently standing on, but on one of the eight neighbouring patches with the lowest creativity score (*cScore*; Table 1.1). This simple form of retention ensures that more favoured patchworks remain longer in the domain ("survival of the fittest enough"). This is the time when patches' state variables *pR*, *pG*, *pB*, *pcolor*, *pDom*, *plabel*, *hue*, *madeBy* and *cScore* are updated.

rate-patchwork

Evaluators rate patchworks according to their perceived novelty and/or appropriateness, and they can be one of three types: *En* (judges only novelty), *Ea* (judges only appropriateness) or *Ena* (judges both). An evaluator's judgement results in one dichotomous creativity score (*cScore*) for a particular patch: 0 = *not creative*, 1 = *creative*. Evaluators judging patchworks based on their perceived novelty do so according to the variable threshold *novelty-stringency*, and those judging appropriateness are influenced by the variable threshold *appropriateness-stringency*. Both thresholds are set by the observer in NetLogo's Interface Tab (see also Table 1.2), so they are the same for all evaluators affected by them. The pseudo code in Figure 1.3 describes how the different evaluator types make their judgements. Note that evaluator type *Ena* must judge a patchwork's novelty as given (1) *and* its appropriateness as given (1) in order to rate the entire patchwork as creative (1). The observer can set how many evaluators of each type are desired by using input boxes in NetLogo's Interface Tab.

Assessing novelty The variable *novelty-stringency* ranges from $0-1$ and represents a threshold according to which an evaluator decides whether a patch under current evaluation is novel or not. The patchwork's (p) relative frequency (rf_{pe}) is calculated from an evaluator's (e) memory, i.e. how often has evaluator e seen patchwork p before compared with all other patchworks it has also seen? If rf_{pe} is less than or equal to *novelty-stringency*, the patchwork is considered novel ($n = 1$), otherwise it is not ($n = 0$; see also pseudo code in Figure 1.3). This way of operationalizing novelty assessments may best be described by the following question: *How unfamiliar*

[1]So creators receive *info-rate* $*$ 3 pieces of information, whereas evaluators receive *info-rate* $*$ 1 pieces of information. Each memory variable (*cR*, *cG*, *cB* and *eMem*) receives *info-rate* pieces of information.

```
; How En and Ena assess a patchwork's novelty:
LET p = relative frequency of current patchwork in my memory
IF p <= novelty-stringency
    THEN  SET novelty = 1 ; novel
    ELSE  SET novelty = 0  ; not novel
; How Ea and Ena assess a patchwork's  appropriateness:
LET h = patchwork's hue ; warm vs. cool
LET viewed = 6 randomly selected neighbors
LET  c = number of viewed with hue = h
SET c = c / 6
IF c >= appropriateness-stringency
    THEN  SET appropriateness  = 1 ; appropriate
    ELSE  SET appropriateness = 0 ; not appropriate
; How a patchwork's creativity score is set:
IF type = En THEN SET cScore = novelty
IF type = Ea THEN SET cScore = appropriatess
IF type = Ena THEN IF (novelty + appropriateness) = 2
    SET cScore = 1 ; creative
    ELSE  SET cScore = 0 ; not creative
```

Figure 1.3: *The pseudo code explains how evaluators En, Ea and Ena judge novelty and appropriateness as well as turn these ratings into a creativity score.*

must a patchwork be to an evaluator to be considered novel? Note that novelty assessment depends an evaluator's memory; two evaluators may disagree about the same patchwork's novelty depending on what they have seen before.

Assessing appropriateness All patches have a variable called *hue*, which indicates whether their colour is warm or cool (see Table 1.1). A patchwork is considered appropriate when its hue agrees with the hues of its neighbours, meaning it should have the same hue as the majority of surrounding patches. An evaluator about to judge a patchwork's appropriateness therefore first takes a look at the latter's surroundings. To simulate imperfection of humans' perception of their environment (see for instance Chattoe, 1998) and to enhance variability of judgements, an evaluator only checks the hues of six randomly selected neighbours[2]. It then assesses how many (c) of the six have the same hue as the patchwork under current evaluation. By calculating the percentage of six neighbours with the same hue ($\frac{c}{6}$), a number from $0 - 1$ results. This number is finally compared with the variable *appropriateness-stringency*. The latter also ranges from $0 - 1$ and represents a threshold according

[2]Without this stochasticity appropriateness judgements would always be the same, as all evaluators would view the same eight neighbours and come to the same judgement according to the algorithm to follow.

to which an evaluator decides whether a patchwork under current evaluation is appropriate or not. If $\frac{c}{6}$ is greater than or equal to *appropriateness-stringency*, the patchwork is considered appropriate. This way of operationalizing appropriateness assessments may best be described by the following question: *At least how much does a patchwork need to fit to its surroundings in order to be considered appropriate by an evaluator?* See the pseudo code in Figure 1.3 for another description.

move

Different movement strategies are available for creators and evaluators, and they can be set by the modeller with a so-called *chooser* in NetLogo's Interface Tab. The strategies were already explained in CRESY-I's ODD Protocol.

forget-some-info

Creators and evaluators forget according to the same-named submodel in CRESY-I's ODD Protocol. Analogous to the way creators forget in CRESY-I, evaluators also forget three randomly selected pieces of information which are deleted from their memory *eMem*. Creators' state variables *cR*, *cG*, and *cB* and evaluators' state variable *eMem* (Table 1.2) are updated in this procedure.

tick

In this submodel, NetLogo's built-in time counter advances. Time is modelled in discrete steps, called *ticks* in NetLogo.

update-domain

During this procedure all dependent variables and the interim variables used for their calculation are updated (Table 1.5). For the exact equations, see Section 1.6 and CRESY-I's ODD Protocol. Additionally, patches' *cScore*s are updated to reflect the evaluation status of the domain. This is also the time when Interface plots are updated.

1.4 Model verification

The following measures were taken to verify CRESY-II:

Coding

- An object-oriented language was used.
- Meaningful variable names were systematically used for all but local variables.
- Sufficient time was allocated for programming.

- A second programmer was asked to proofread parts of CRESY-II.
- Assertions were added to check whether input parameter values made sense.
- NetLogo's syntax analysis button ("check" in the Procedures Tab; Wilensky, 1999) was regularly used.
- Additional pieces of code were added to each procedure to check input and output plausibility. This code was deleted in the final version.

Documenting

- Comments were added to the code to facilitate model-to-program translation.
- NetLogo's Information Tab (Wilensky, 1999) and the ODD protocol (Grimm et al., 2006) were used.

Observing

- An abundance of output diagnostics (histograms, monitors, plots of interim and final variables) were programmed in the NetLogo's Interface Tab (Wilensky, 1999).
- The model was animated using NetLogo's "View" (Wilensky, 1999).
- During model building, individual procedures were run singly and their output checked in NetLogo's Command (Wilensky, 1999).
- During model building, an abundance of brief experiments were carried out to check if CRESY-II functions just as CRESY-I does in the procedures carried over from the latter model.

Testing

- Before experimentation, conditions for which output parameters are known were tested by observation. CRESY-II was not used until the values were as expected.
- All experiments conducted with CRESY-II were run with (theoretically derived) extreme conditions.
- In all experiments conducted with CRESY-II, graphical and statistical testing was used to back up modeller observations.

Comparing

- After each experiment conducted with CRESY-II, data interpretations and discussions containing model-to-program comparisons were documented.

1.5 Independent variables

Figure 1.4 depicts a chart containing all parameters an experimenter can vary in CRESY-II. They are classified according to the system they belong to. Method-specific variables result due to the peculiarities of simulation modelling. The modeller can influence the size and shape of the programmed world. Furthermore, the number of runs and steps per run require setting for experimentation. The macrosystem and creator microsystem variables are the same as used in CRESY-I. Technical variables in the evaluator microsystem correspond to those in the creator microsystem: The number of evaluators can be altered, and their movement strategies and information uptake rate may be varied. Substantial variables in this system are the ratio of evaluator types (*eRatio*), their novelty and appropriateness thresholds (*novelty-stringency*, *appropriateness-stringency*) and the number of categories they have to perceive patchworks (*domSize*). Note that *domSize* was a global variable in CRESY-I. In CRESY-II evaluation takes place in evaluators as autonomous agents and not in one global observer. Therefore, *domSize* now belongs to the former. It is set to one value for all evaluators.

In the experiments conducted with CRESY-II, not all parameters in Figure 1.4 are varied due to reasons of feasibility. Note that if each variable was varied in two ways, resulting in "simple" a 2^k design, a total of $524,288$ experimental conditions would be necessary. In the documentation of each experiment conducted with CRESY-II, the independent variables chosen for experimentation are explicitly stated.

1.6 Dependent variables

1.6.1 hCAII

This dependent variable was used in CRESY-I and is reused in CRESY-II. It's formula is thoroughly explained in CRESY-I's ODD Protocol.

1.6.2 hX.Y

This dependent variable was used in CRESY-I and is reused in CRESY-II. It's formula is thoroughly explained in CRESY-I's ODD Protocol.

1.6.3 Creativity

After rating a patchwork's novelty and/or appropriateness, evaluators ultimately judge its creativity by giving it a binary score (*cScore*) of 0 (not creative) or 1 (creative). Each evaluator rates every patchwork made in the current step. So *number-creators* $*$ *number-evaluators* creativity scores result every tick, because each creator makes only one patchwork and each evaluator rates every one. Subsequently, a total creativity assessment for the current step is calculated and expressed in the variable

Method-specific Variables	
Substantial	**Technical**
	number of runs number of steps (ticks) world size world shape (e.g. borders)
Macrosystem Variables	
Substantial	**Technical**
predefine-r predefine-b predefine-g	
Creator Microsystem Variables	
Substantial	**Technical**
cRatio imagination	number-creators info-rate movement
Evaluator Microsystem Variables	
Substantial	**Technical**
eRatio domSize novelty-stringency appropriateness-stringency	number-evaluators info-rate number-evaluators

Figure 1.4: *The input parameters available in CRESY-II are divided into method-specific, macrosystem and microsystem variables. They are subdivided into technical and substantial variables.*

crea. For each patchwork j currently evaluated, its number of corresponding creative evaluations (1s; $N_{1\,j}$) is set in relation to its total number evaluations ($N_{total\,j}$, equal to *number-evaluators*). That is the single patchworks's current creativity score (c_j). These interim creativity scores are then averaged to obtain the patchworks' mean creativity in the current step.

$$crea = \frac{\sum_{i=1}^{n} c_j}{n} \tag{1.1}$$

whereas n = number of patchworks made in the current step (equal to *number-creators*),

and $c_j = \frac{N_{1\,j}}{N_{total\,j}}$ for patchwork j

crea is therefore a percentage indicating how creative evaluators find creators' current work on average. Note again that *crea* is calculated every step only with the patchworks currently made and thus evaluated.

1.6.4 Reliability

Regardless of output type, measuring the accuracy or reliability of judgements is an essential part of creativity assessment (Silvia, 2008). In CRESY-II, reliability is measured with Fleiss' κ (Fleiss, 1971) for binary data as derived in (Bortz et al., 2008, p. 454-458).

1.7 Experimental design

Two experiments were conducted with CRESY-II (see Figure 1.5). The first, CII-1a, was conceived to investigate how many runs, steps per run and evaluators (*number-evaluators*) should be used for further explorations. The results obtained were used to design the second experiment, CII-1b. It's purpose was to explore how macrosystem, creator microsystem and evaluator microsystem variables affect the dependent variables. This experiment was therefore designed to investigate substantial variables regarded as essential in Csikszentmihalyi's (1999) model of creativity. The results gathered were used to generate hypotheses about his perspective. Note that fewer preparatory experiments were conducted with CRESY-II compared with CRESY-I. This is due to the fact that only parameters values already used in CRESY-I were reused in CRESY-II, i.e. input parameters available in both models are set in CRESY-II with values already tested in CRESY-I. For more on these experiments, please contact the author.

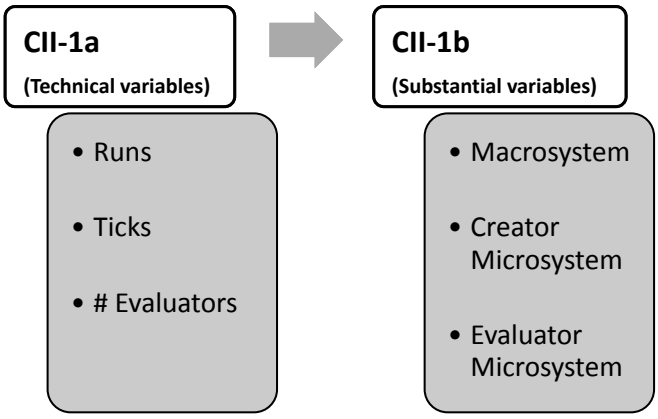

Figure 1.5: *Overview of experiments conducted with CRESY-II.*

Bibliography

Amabile, T. (1996). *Creativity in context*. Boulder, CO: Westview Press.

Amabile, T. & Mueller, J. (2007). Psychometric approaches to the study of human creativity. In J. Zhou & C. Shalley (Eds.), *Handbook of organizational creativity* (pp. 33–64). New York: Lawrence Erlbaum Associates.

Bechtoldt, M., De Dreu, C., & Choi, H. (2010). Motivated information processing, social tuning, and group creativity. *Journal of Personality and Social Psychology*, 99(4), 622–637.

Bortz, J., Lienert, G., & Boehnke, K. (2008). *Verteilungsfreie Methoden in der Biostatistik [Non-parametric methods in biostatistics]*. Heidelberg, Germany: Springer Medizin Verlag.

Chattoe, E. (1998). Just how (un)realistic are evolutionary algorithms as representations of social processes? *Journal of Artificial Societies and Social Simulation*, 1(3), 2.

Csikszentmihalyi, M. (1988). Society, culture, and person: A systems view of creativity. In R. Sternberg (Ed.), *The nature of creativity. Contemporary psychological perspectives* (pp. 325–339). Cambridge, UK: Cambridge University Press.

Csikszentmihalyi, M. (1999). Implications of a systems perspective for the study of creativity. In R. Sternberg (Ed.), *Handbook of creativity* (pp. 313–338). Cambridge, UK: Cambridge University Press.

Dennis, A. & Williams, M. (2003). Electronic brainstorming. In P. Paulus & B. Nijstad (Eds.), *Group creativity* (pp. 160–178). New York: Oxford University Press.

Fleiss, J. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5), 378–382.

Ford, C. & Kuenzi, M. (2007). Organizing creativity research through historical analyses of foundational administrative science texts. In J. Zhou & C. Shalley (Eds.), *Handbook of organizational creativity* (pp. 65–94). New York: Lawrence Erlbaum Associates.

Grimm, V., Berger, U., Bastiansen, F., Eliassen, S., Ginot, V., Giske, J., Goss-Custard, J., Grand, T., Heinz, S., Huse, G., Huth, A., Jepsen, J., Jorgensen, C., Mooij, W., Müller, B., Pe'er, G., Piou, C., Railsback, S., Robbins, A., Robbins, M., Rossmanith, E., Rüger, N., Strand, E., Souissi, S., Stillman, R., Vabo, R., Visser, U., & DeAngelis, D. (2006). A standard protocol for describing individual-based and agent-based models. *Ecological Modelling*, 198, 115–126.

Grimm, V., Berger, U., DeAngelis, D., Polhill, G., Giske, J., & Railsback, S. (2010). The ODD protocol: A review and first update. *Ecological Modelling*, 221, 2760–2768.

Janssen, M., Alessa, L., Barton, M., Bergin, S., & Lee, A. (2008). Towards a community framework for agent-based modelling. *Journal of Artificial Societies and Social Simulation*, 11(2), 6.

Kahl, C. (2009). Modeling a social environment to investigate creativity: An agent-based model. In E. Witte & C. Kahl (Eds.), *Sozialpsychologie der Kreativität und Innovation [Social psychology of creativity and innovation]* (pp. 27–52). Lengerich, Germany: Pabst Science Publishers.

Kaufman, J., Baer, J., Cole, J., & Sexton, J. (2008). A comparision of expert and nonexpert raters using the consensual assessment technique. *Creativity Research Journal*, 20(2), 171–178.

Kozbelt, A. & Serafin, J. (2009). Dynamic evaluation of high- and low-creativity drawings by artist and nonartist raters. *Creativity Research Journal*, 21(4), 349–360.

Lonergan, D., Scott, G., & Mumford, M. (2004). Evaluative aspects of creative thought: Effects of appraisal and revision standards. *Creativity Research Journal*, 16(2), 231–246.

Lubart, T. (1999). Creativity across cultures. In R. Sternberg (Ed.), *Handbook of creativity* (pp. 339–350). Cambridge, UK: Cambridge University Press.

Metzger, W. (1979/1986). Gestalttheoretische Ansätze zur Fragen der Kreativität [Gestalt psychology approaches to creativity]. In *Gestalt-Psychologie [Gestalt psychology]* (pp. 522–528). Frankfurt a.M., Germany: Kramer.

Müller, C. (2009). Hier ist es schwer, verdammt schwer [It's difficult here, extremely difficult]. *sonntaz (14 Aug 2009)*, retrieved March 10, 2011, from [http://www.taz.de/1/leben/kuenste/artikel/1/hier-ist-es-schwer-verdammt-schwer/].

Ochse, R. (1990). *Before the gates of excellence. The determinants of creative genius.* Cambridge, UK: Cambridge University Press.

Polhill, G., Galan-Diaz, C., Gotts, N., Craig, T., Marshall, K., Sutherland, L., Kriel, A., & Fischer, A. (2010). The ODDness of modelling: Early experiences from a transdisciplinary modelling exercise. In A. Ernst & S. Kuhn (Eds.), *Proceedings of the 3rd World Congress on Social Simulation WCSS2010* (pp. 1–8; CD–ROM). Kassel, Germany: Center for Environmental Systems Research, University of Kassel.

Polhill, J., Parker, D., Brown, D., & Grimm, V. (2008). Using the ODD protocol for describing three agent-based social simulation models of land-use change. *Journal of Artificial Societies and Social Simulation*, 11(23), 30.

Preiser, W. (2006). Creativity research in German-speaking countries. In J. Kaufman & R. Sternberg (Eds.), *The international handbook of creativity* (pp. 167–201). Cambridge, UK: Cambridge University Press.

Rietzschel, E., Nijstad, B., & Stroebe, W. (2006). Productivity is not enough: A comparison of interactive and nominal brainstorming groups on idea generation and selection. *Journal of Experimental Social Psychology*, 42, 244–251.

Rigney, D. (2001). *The metaphorical society: An invitation to social theory*. Lanham, MD: Rowman & Littlefield.

Runco, M., McCarthy, K., & Svenson, E. (1994). Judgments of the creativity of artwork from students and professional artists. *The Journal of Psychology*, 128(1), 23–31.

Sawyer, R. (2006). *Explaining creativity. The science of human innovation*. New York: Oxford University Press.

Schuler, H. & Görlich, Y. (2006). *Kreativität: Ursachen, Messung, Förderung und Umsetzung in Innovation [Creativity: Causes, Assessment, Promotion and Transformation to Innovation]*. Göttingen, Germany: Hogrefe.

Shannon, C. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27, 379–423, 623–656.

Silvia, P. (2008). Discernment and creativity: How well can people identify their most creative ideas? *Psychology of Aesthetics, Creativity, and the Arts*, 2(3), 139–146.

Sternberg, R. (1999). *Handbook of creativity*. Cambridge, UK: Cambridge University Press.

Sternberg, R., Grigorenko, E., & Singer, J., Eds. (2004). *Creativity. From potential to realization*. Washington, D.C.: American Psychological Association.

Stokes, P. (1999). Learned variability levels: Implications for creativity. *Creativity Research Journal*, 12(1), 37–45.

Stokes, P. (2007). Using constraints to generate and sustain novelty. *Psychology of Aesthetics, Creativity, and the Arts*, 1(2), 107–113.

Styhre, A. (2006). Organization creativity and the empiricist image of novelty. *Creativity and Innovation Management*, 15(2), 143–149.

van den Besselaar, P. & Leydesdorff, L. (2009). Past performance, peer review, and project selection: A case study in the social and behavioral sciences. *Research Evaluation*, 21(4), 273–288.

Ward, T., Smith, S., & Finke, R. (1999). Creative cognition. In R. Sternberg (Ed.), *Handbook of creativity* (pp. 189–212). Cambridge, UK: Cambridge University Press.

Wilensky, U. (1999). *Netlogo*. Evanston, IL: Center for Connected Learning and Computer-Based Modeling, Northwestern University. Retrieved October 20, 2010, from [http://ccl.northwestern.edu/netlogo/].

Zhou, J. & Shalley, C., Eds. (2007). *Handbook of organizational creativity*. New York: Lawrence Erlbaum Associates.

Zysno, P. & Bosse, A. (2009). Was macht Gruppen kreativ? [What makes groups creative?]. In E. Witte & C. Kahl (Eds.), *Sozialpsychologie der Kreativität und Innovation [The social psychology of creativity and innovation]* (pp. 120–150). Lengrich, Germany: Pabst Science Publishers.

Acknowledgements

I find that all creative work is collaborate in some manner. Therefore...

I greatly appreciate the valuable suggestions made by the following researchers regarding this model or its earlier versions: Hans Hansen, Erich H. Witte, Klaus Troitzsch, Nigel Gilbert, Iris Lorscheid, Nanda Wijermans, the SIMSOC@WORK group, Florian Felix Meyer, Mario Paolucci. Thank you to the following scientists for asking particularly helpful questions about CRESY-I: Rosaria Conte, Edmund Chattoe-Brown.

Many thanks to Seth Tisue, James Steiner, Nick Bennett and sanbalestrini from the NetLogo Users' Group for their fast and helpful replies to my posts as well as for their suggested pieces of code.

I would also like to credit the participation I received from approximately 40 students with a semester's background in creativity research who "played" with CRESY-I in January 2011 and created hypotheses about the model based on their observations. Thanks to their confusion, the Interface now contains more explanatory notes!

Last but definitely not least thanks to Uri Wilensky for NetLogo and the reusable code in models from the NetLogo Models Library!