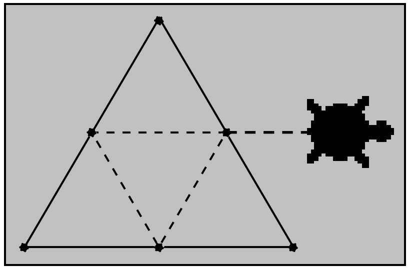


FEEC
Faculdade de Engenharia
Elétrica e de Computação
UNICAMP

Development of Software for Educational Activities

Leo Burd

DCA/FEE/UNICAMP

(Based on my MSc's thesis defense - October, 1999)

The problem

There is a lack of specific methodologies targeted to educational software development

Goals

- Propose an approach to educational software development based on Activity Theory

- Analyse the application of the approach on constructionist education and, more specifically, on a practical activity using the Cocoa software

Key ideas

- No given software or technology is, by itself, able to educate a person.

- If we believe that education is the result of a set activities that create favorable conditions for teaching and learning, then computational systems will be more or less “educational” depending on the kind of support that they offer to those activities.

Hypothesis

- Activity Theory can help describe important aspects of educational activities and guide the development of software for education

Points to consider

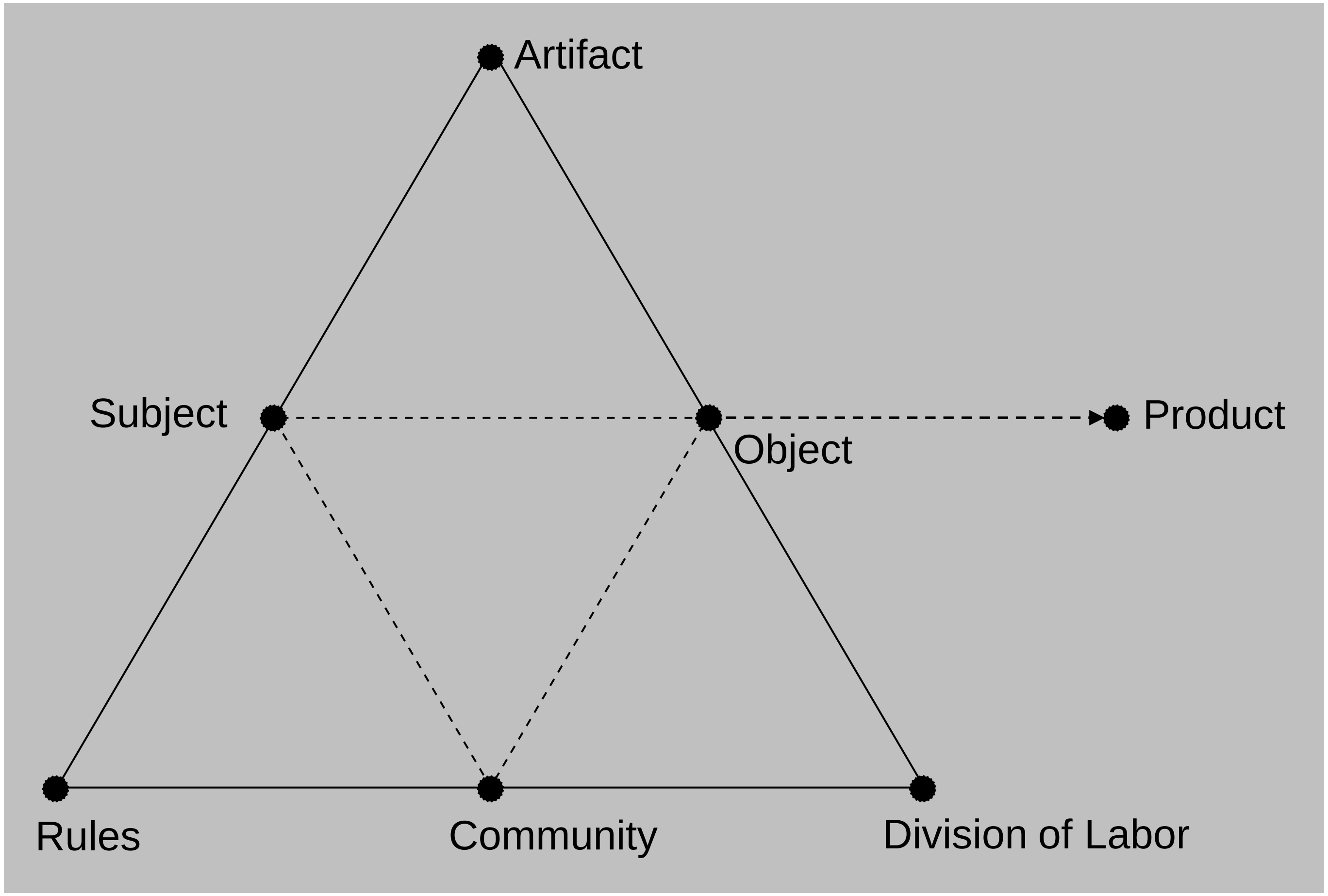
- What is understood by “educational activity”?
- How to describe it for educators and software engineers?
- What is the specific role of computers within the activity?

Activity Theory

- Theoretical body that studies human activities and, in special, the sociocultural dynamics and the importance of tools within this context
- Is gaining space in education and in the area of software engineering that deals with human-computer interaction (HCI)

The activity diagram

(based on Engeström’s)



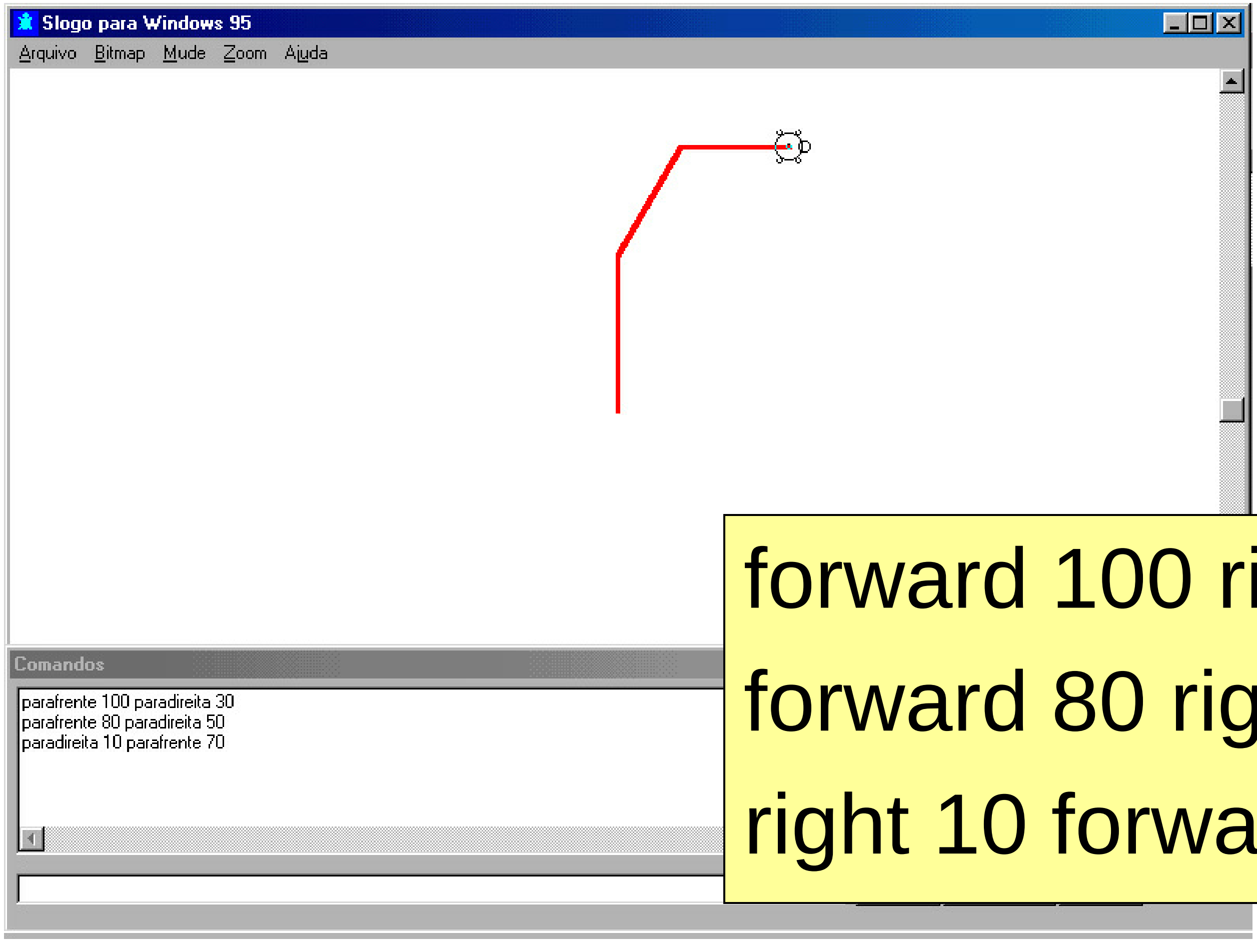
Aspects analyzed by Activity Theory

- Hierarchical structure of the activity
- Internalization and externalization
- Object-orientedness
- History and development
- Mediation

Constructionism

- Theory that studies the use of technology in the creation of educational environments
- Developed in the early 1970s by Seymour Papert
- Most well-known by Logo, an education software used by millions of people all over the world

Logo’s turtle geometry



```
parafrente 100 paradireita 30
parafrente 80 paradireita 50
paradireita 10 parafrente 70
```

forward 100 right 30

forward 80 right 50

right 10 forward 70

Construcionism

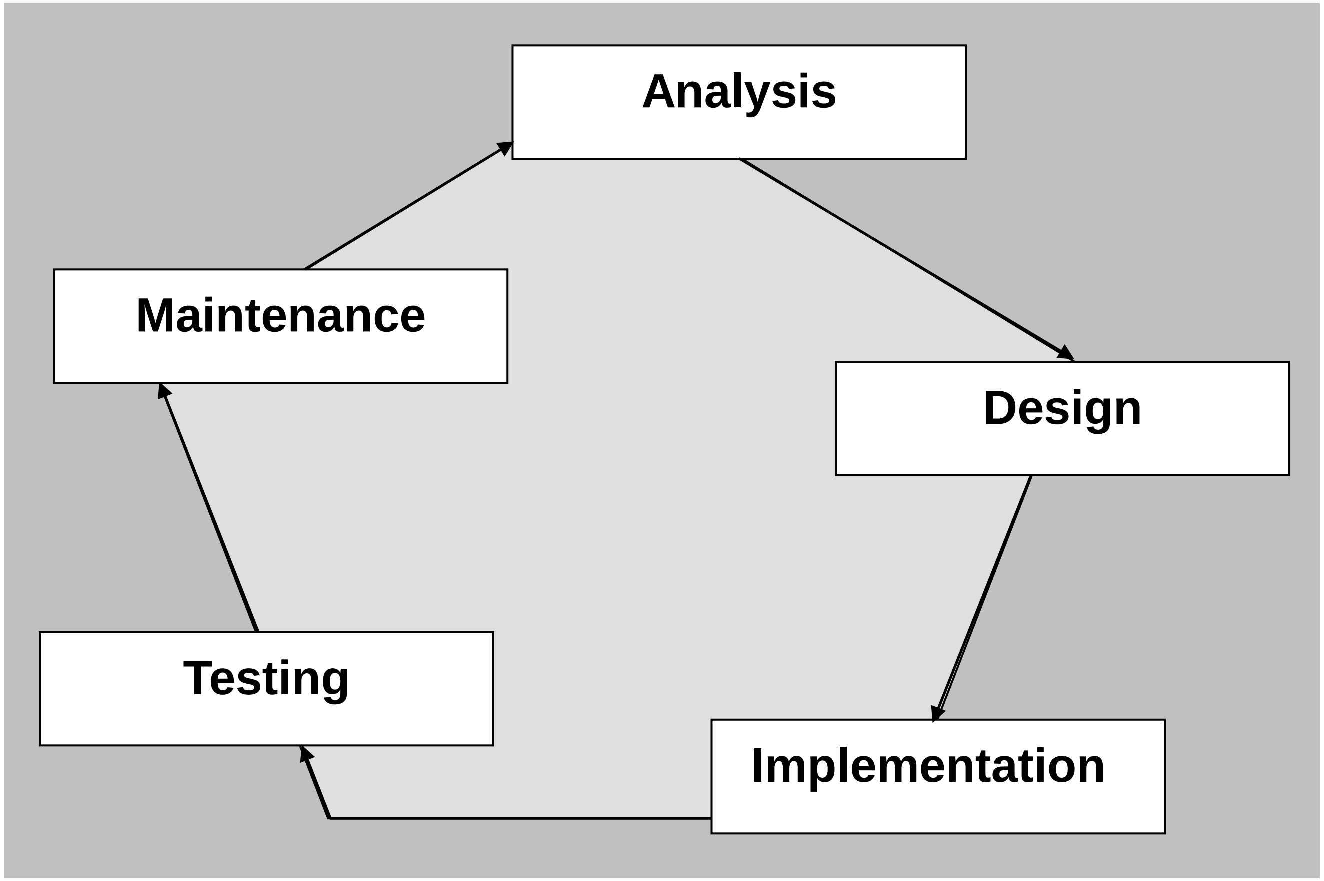
- Problem: there is too much focus on Logo
- The constructionist practice goes beyond the mere use of a tool
- How to develop software to support the constructionist approach of other domains?

There is a lack of methodologies for the development of construtionist software...

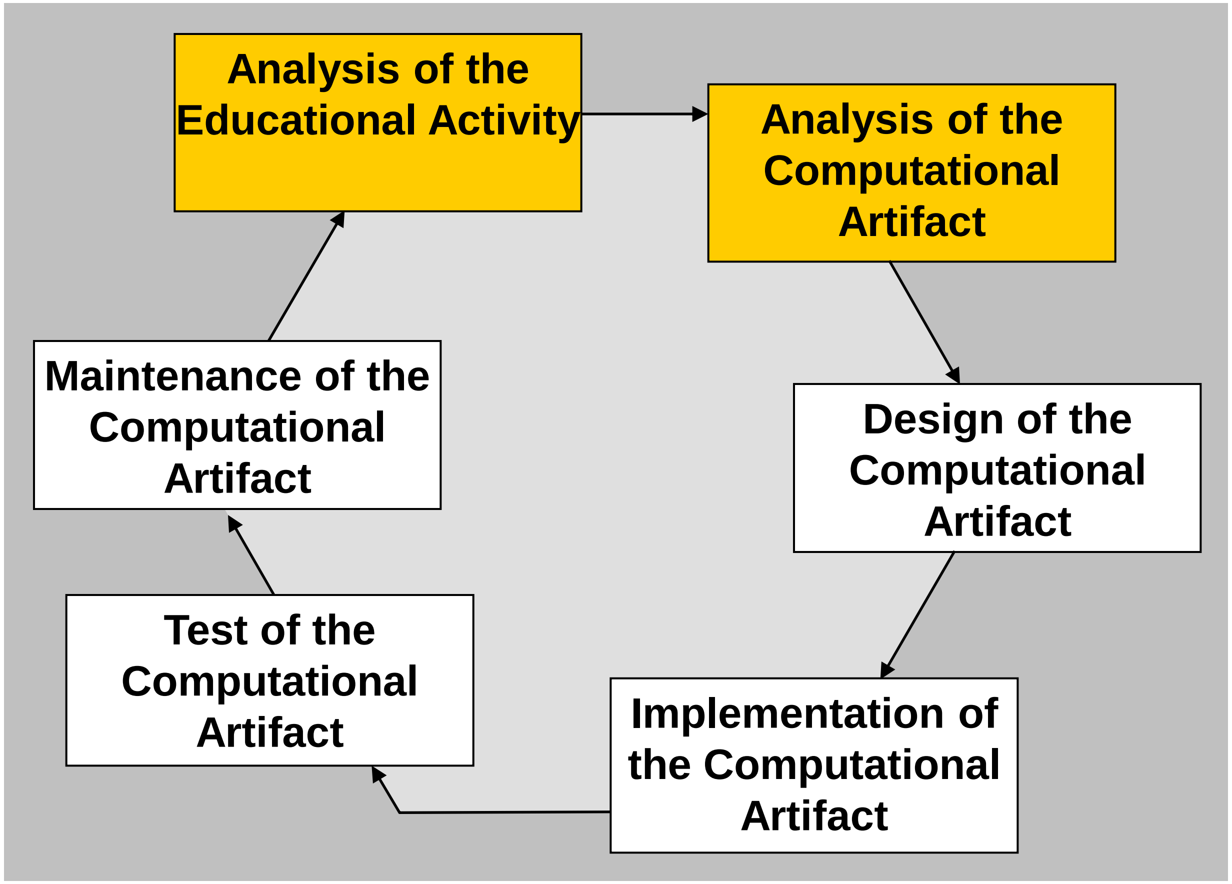
The proposed approach

- Is based on a sociocultural and historical analysis of the educational activity
- Identifies the specific contributions of the computer
- Opens space for the other “more technical” software engineering methodologies

The traditional software development life-cycle



The proposed development cycle

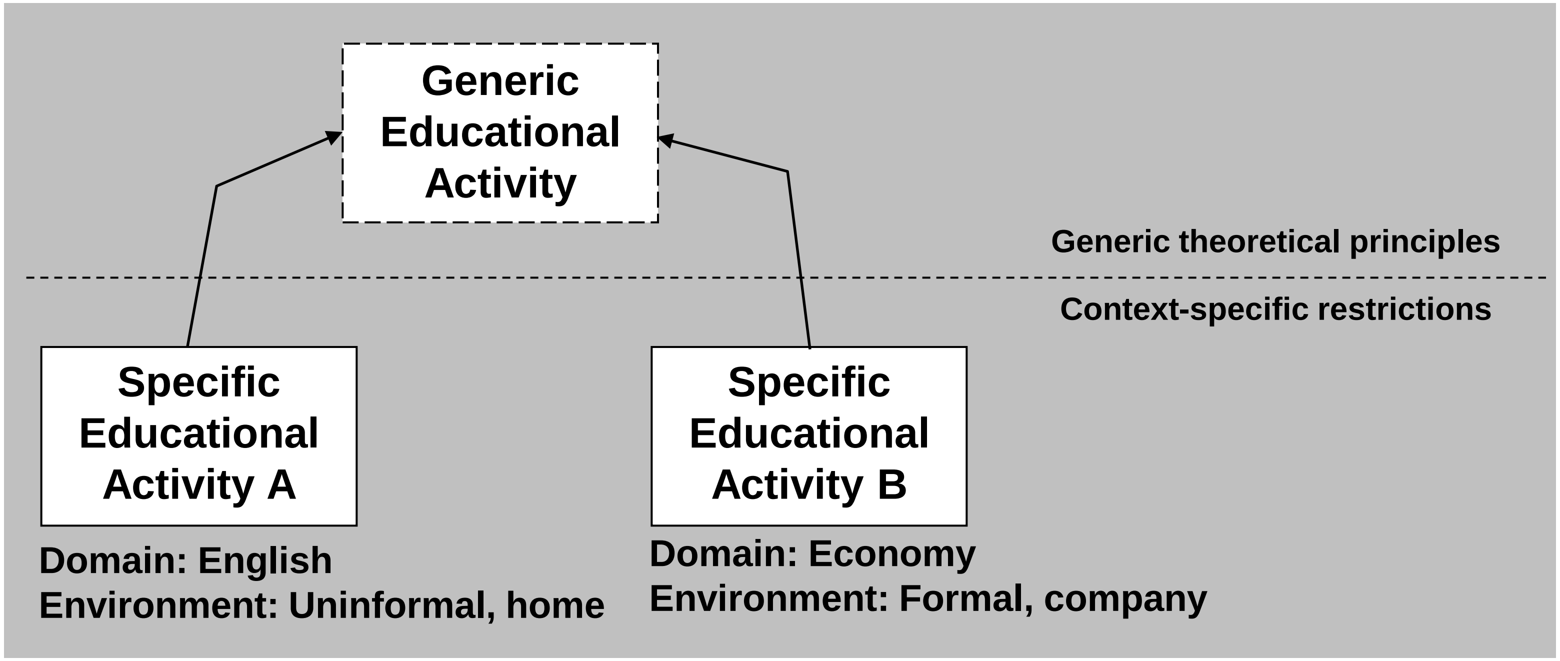


Analysis of the Educational Activity
Analysis of the Computational Artifact
Design of the Computational Artifact
Implementation of the Computational Artifact
Test of the Computational Artifact
Maintenance of the Computational Artifact

The analysis of the educational activity

- Describes the educational activity in all its dimensions
- Is oriented by guiding questions
- Has both a generic and a specific part

The generic and specific parts of the educational activity


Generic Educational Activity
Generic theoretical principles
Context-specific restrictions
Specific Educational Activity A
Specific Educational Activity B

Domain: English
Environment: Uninformal, home

Domain: Economy
Environment: Formal, company

The analysis of the computational artifact

- Describes the role of computers within the activity

- Distinguishes among *computational artifacts, digital artifacts, digital objects and applications*

- Is also driven by guiding questions

“Activity hierarchy” questions

- What is the activity?
- What are the actions that each actor will perform in each role?
- Which actors and roles will the tool support?
- What kinds of operations should be automatized?

“Internalization and externalization” questions

- What concepts, skills, attitudes are to be internalized by each actor?
- What concepts, skills, attitudes are to be externalized by each actor?
- How is the tool going to support that?

“Object orientation” questions

- What’s the object of the activity?
- What are the objects being transformed by each actor?
- What aspects of those objects are the most relevant for each actor?
- How is the tool going to support that?

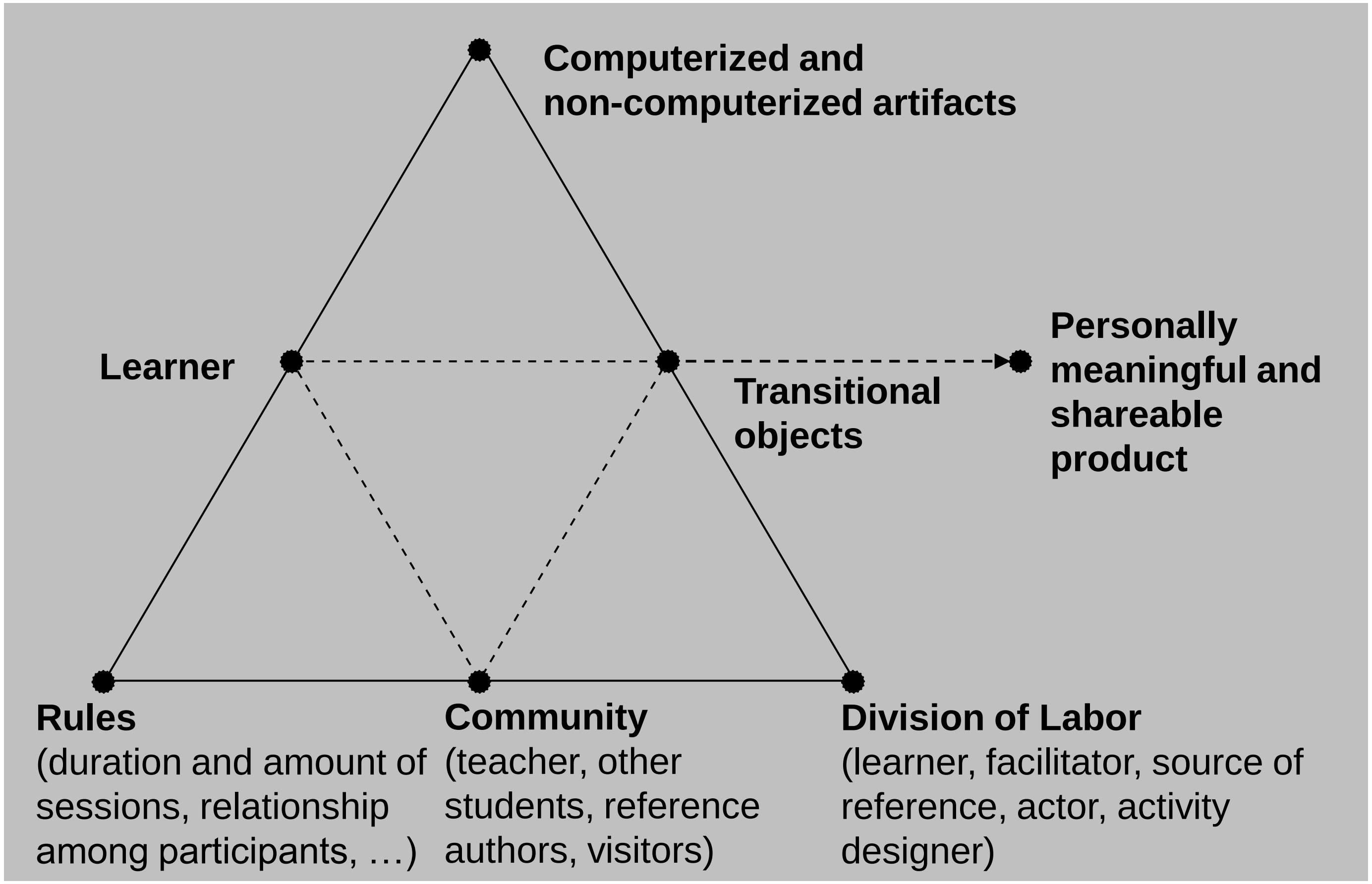
“History and development” questions

- How does the activity evolve over time?
- How do different elements evolve within the activity?
- What historical elements of the environment affect the activity? How?
- How is the tool going to support that?

“Mediation” questions

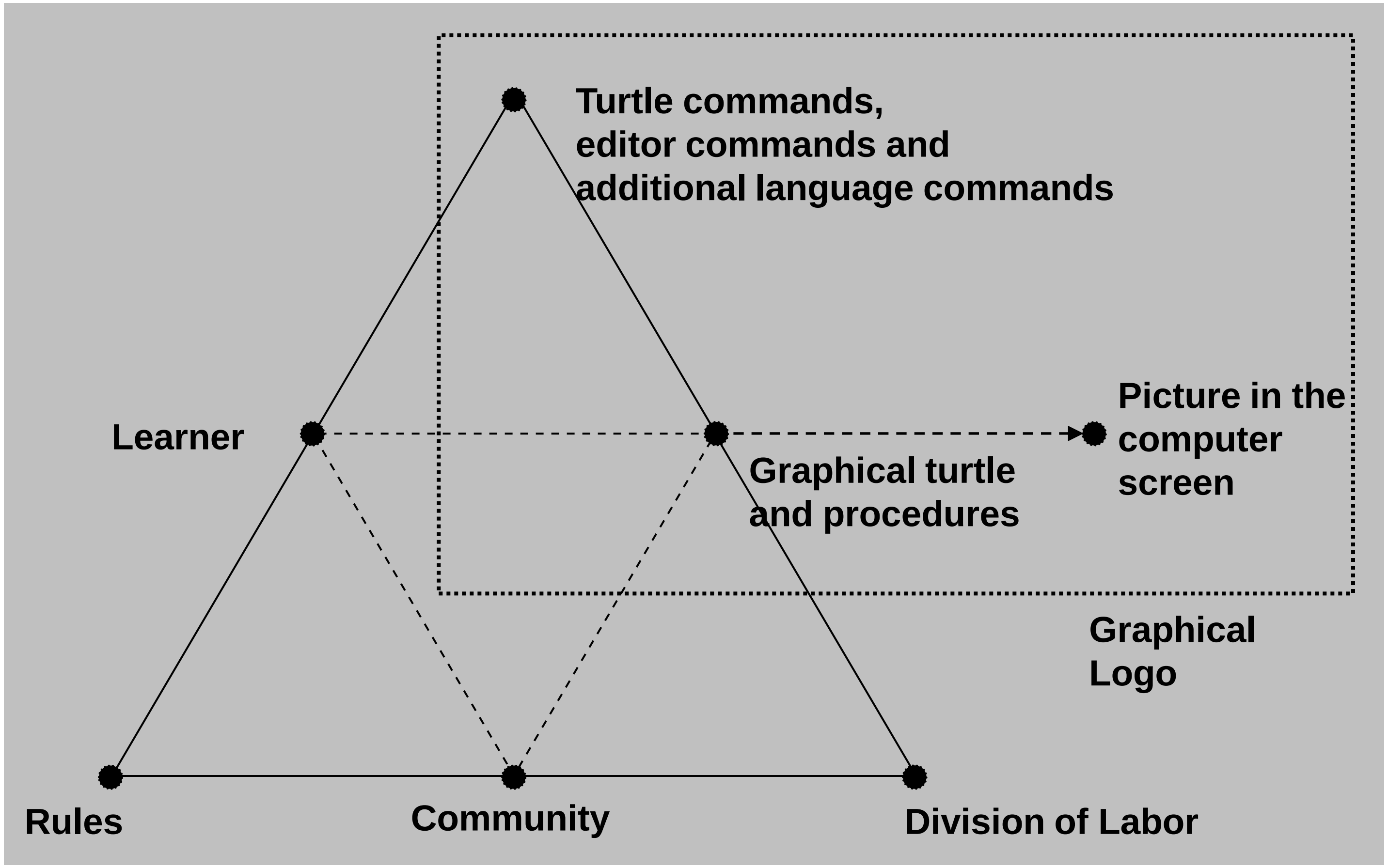
- What elements are going to mediate the interaction between actors and between actors and objects?
- What aspects of the interaction is the tool going to enhance?
- What aspects of the interaction is the tool going to limit?

A Generic Constructionist Learning Activity



Computerized and non-computerized artifacts
Learner
Transitional objects
Personally meaningful and shareable product
Rules
(duration and amount of sessions, relationship among participants, …)
Community
(teacher, other students, reference authors, visitors)
Division of Labor
(learner, facilitator, source of reference, actor, activity designer)

A Graphical Logo Activity



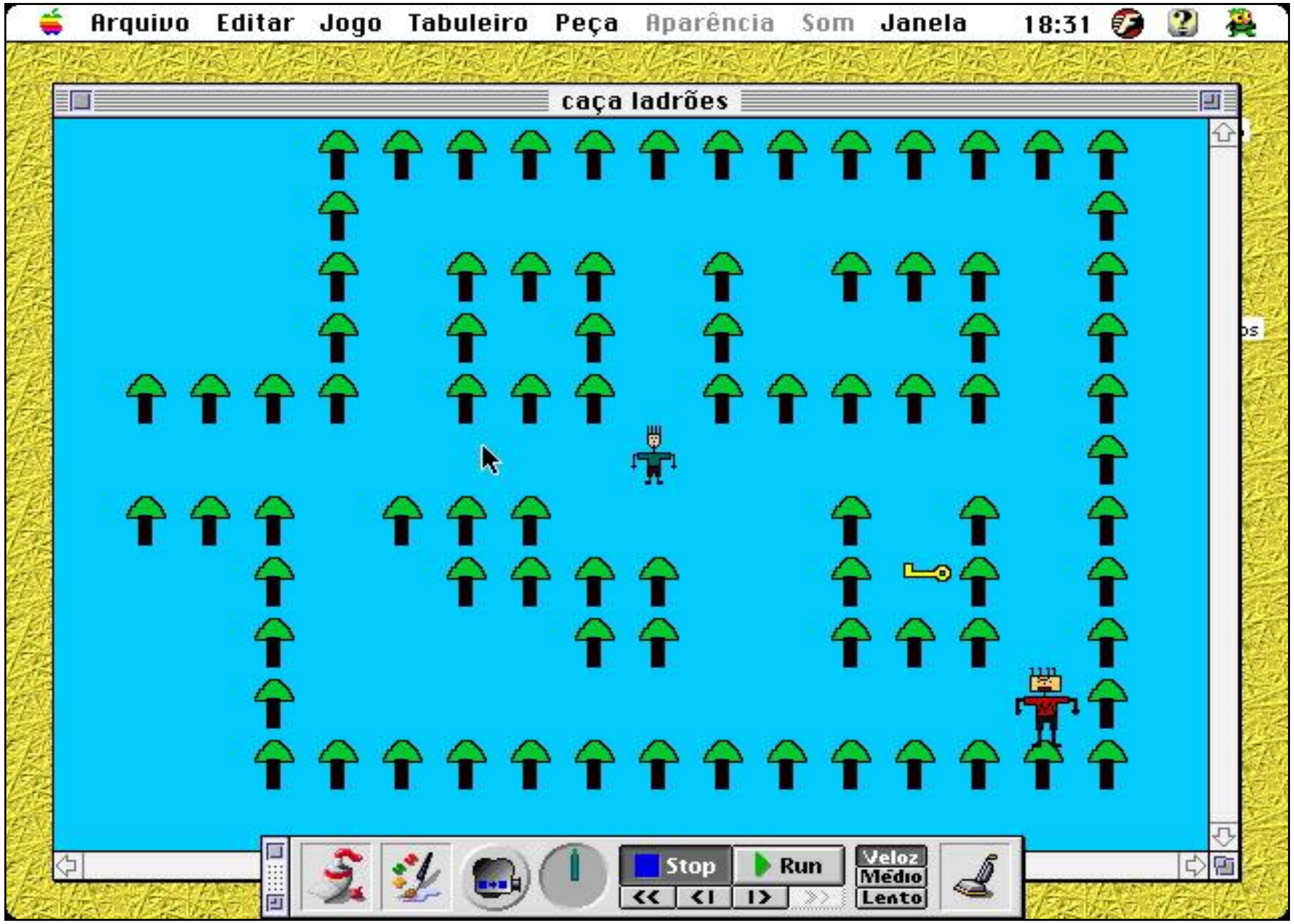
Learner actions

- Idealization
- Construction
- Assessment
- Debugging
- Description

A practical case: The Game’s Workshop

- Institute of the III Millennium - Ibirapuera Park
- “Development of games about the park using the Cocoa software”
- Served as an object to think about this work

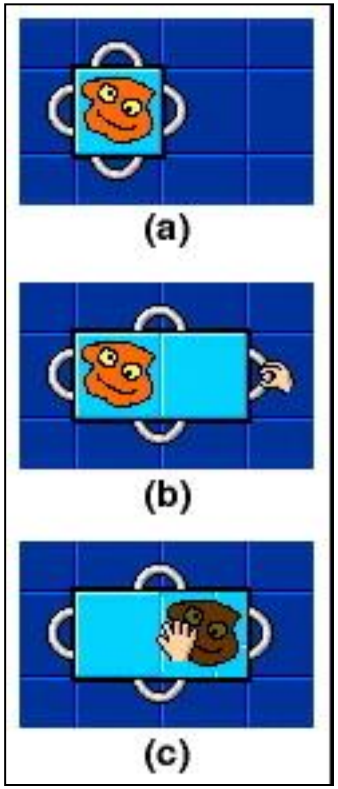
A Cocoa screen



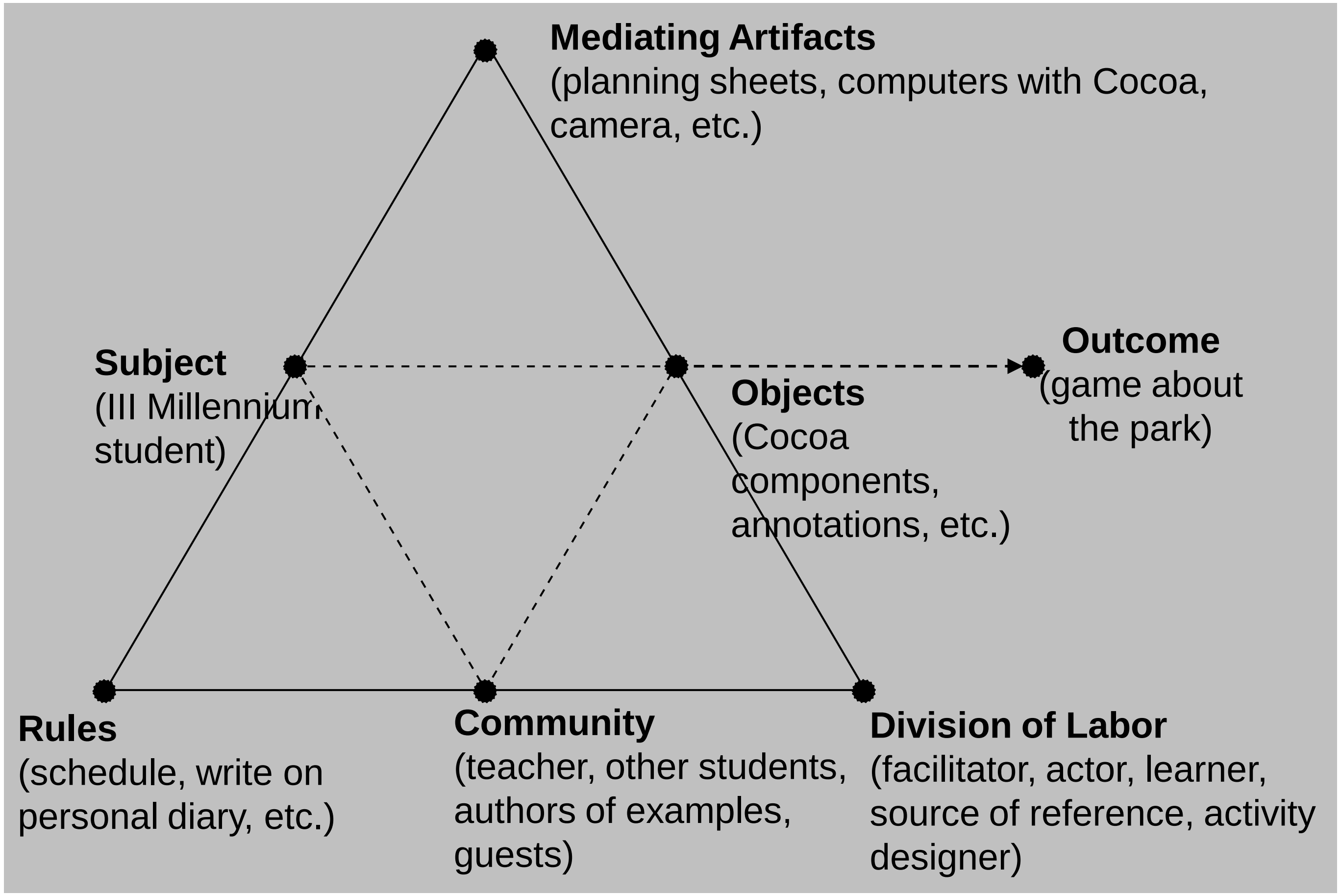
Why Cocoa?

- Considered a successor of the Graphical Logo
  - A programming language for kids
  - Created within the concepts of multimedia, Internet, windows and mouse

- It was relatively easy to adapt it to Portuguese



The Workshop activity



Mediating Artifacts
(planning sheets, computers with Cocoa, camera, etc.)

Subject
(III Millennium student)

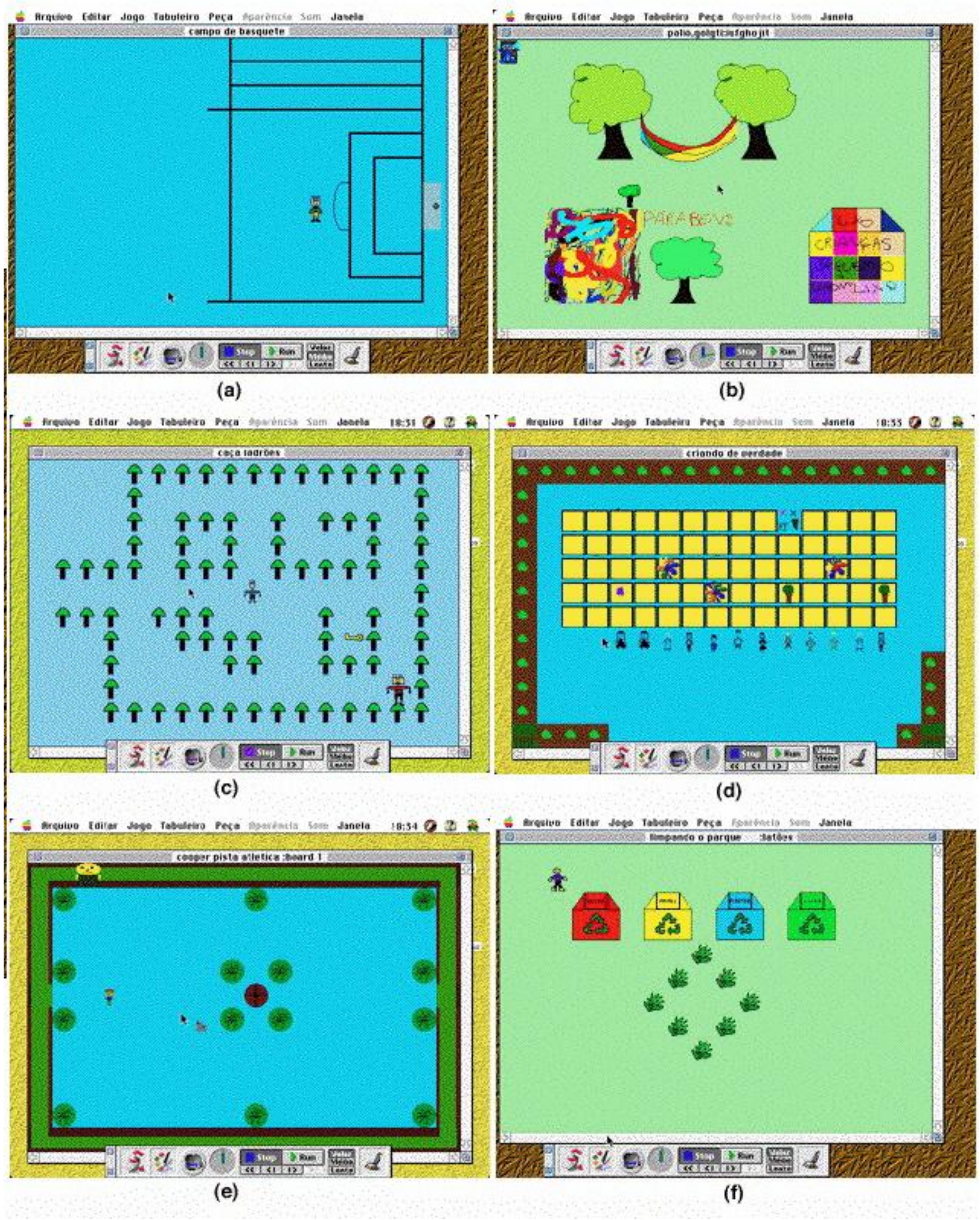
Objects
(Cocoa components, annotations, etc.)

Outcome
(game about the park)

Rules
(schedule, write on personal diary, etc.)

Community
(teacher, other students, authors of examples, guests)

Division of Labor
(facilitator, actor, learner, source of reference, activity designer)


Arquivo Editar Jogo Tabuleiro Peça Aparência Som Janela
campo de basquete
palio.golgi:sfghojit
PARABENS
caça ladrões
criando de verdade
cooper pista atletica :board 1
limpando o parque :latões
Stop
Run

(a) (b)

(c) (d)

(e) (f)

Conclusions from the workshop

- Computers should be seen as components within a larger context
- Applications (such as Cocoa) should be analyzed as part of the computational artifact
- There is a lack of tools to support the facilitator (assessment, annotations)

Conclusions from the workshop

- There is a lack of tools to support the development of long activities (backups, student’s notes)
- It is important to allow time for the appropriation of the artifact (instrumentalization)
- Cocoa could offer tools to avoid student’s distraction and to facilitate the description of student’s actions

Overall research conclusions

- The proposed approach addressed important aspects of the educational activity:
  - Theoretical orientation
  - Specific characteristics of the environment
  - Social dynamics (relationship between roles)
  - Development of the participants
  - Analysis of the computer against and in partnership with other technologies

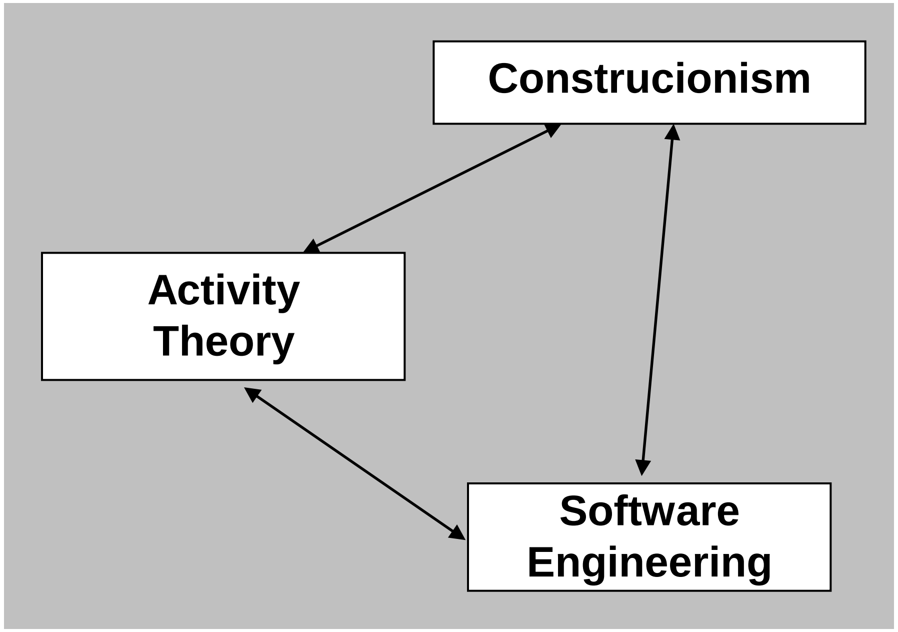
In relation to Constructionism

- Helped organize the theoretical concepts
- Opened space for discussion with other fields of research
- Made explicit Logo characteristics to be explored in other applications
- Raised questions for further research

Questions raised about Constructionism

- Student motivation for action description
- New understanding of computers
- Cycle of idealization-construction-assessment-debugging-description actions
- Importance of the instrumentalization phase
- Lack of tools to support facilitators

Major challenges



- Interdisciplinarity
- Lack of references
- Areas that are recent and in constant evolution
- Translation problems

Major contributions

- Made explicit the need for a methodology for the development of software for education
- Proposed an approach for that
- Reinforced the idea that computers should be analyzed in relation to a context of use
- Critically revised the concepts of Activity Theory and Constructionism
- Articulated the theories studied

Open questions

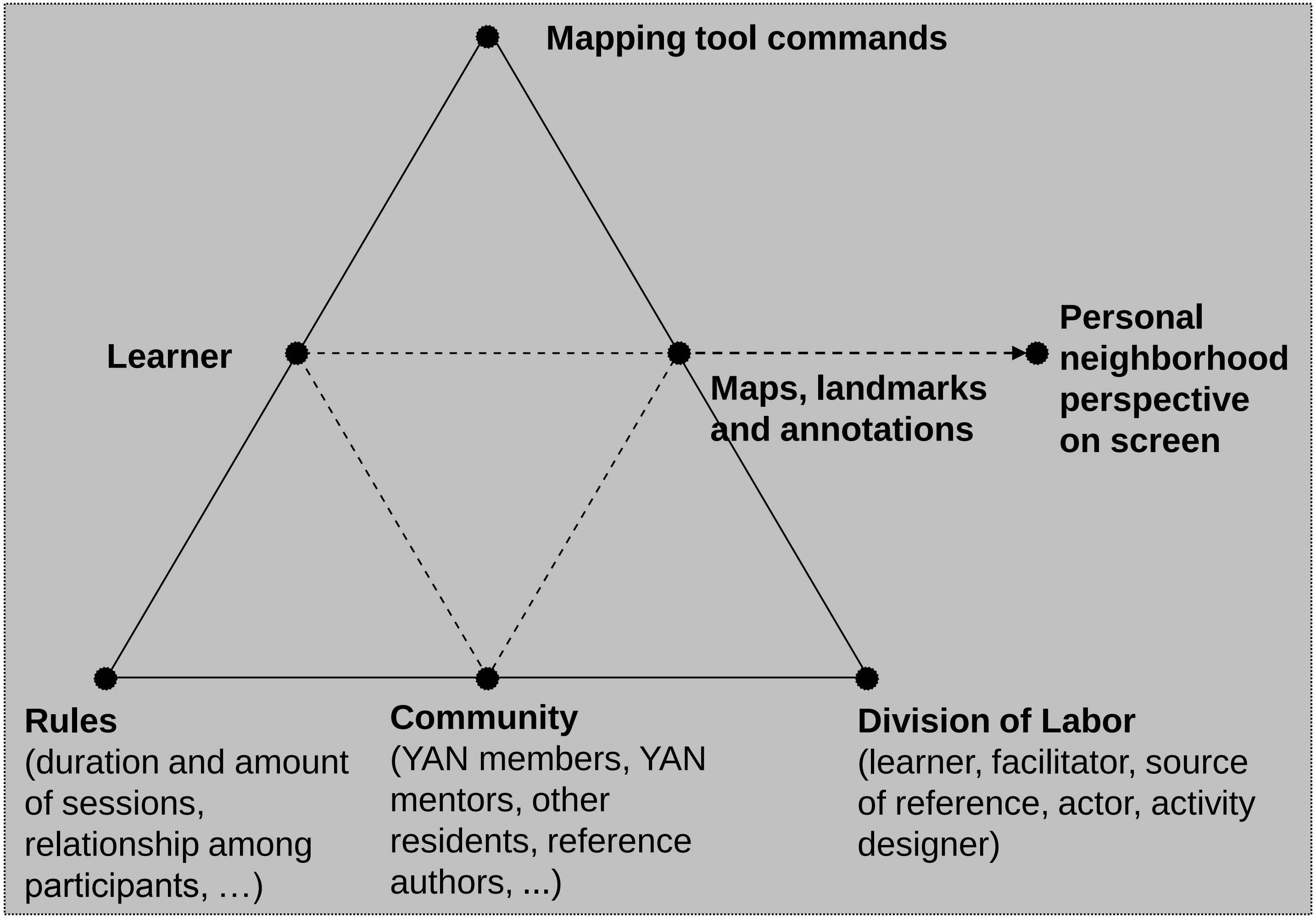
- Can the proposed approach be used with other educational theories?
- Can it be applied to non-education areas?
- How to turn it into a real methodology?
- How to make it more useful for educators and engineers?

Next steps (as of 1999)

- Write articles to spread and improve the ideas

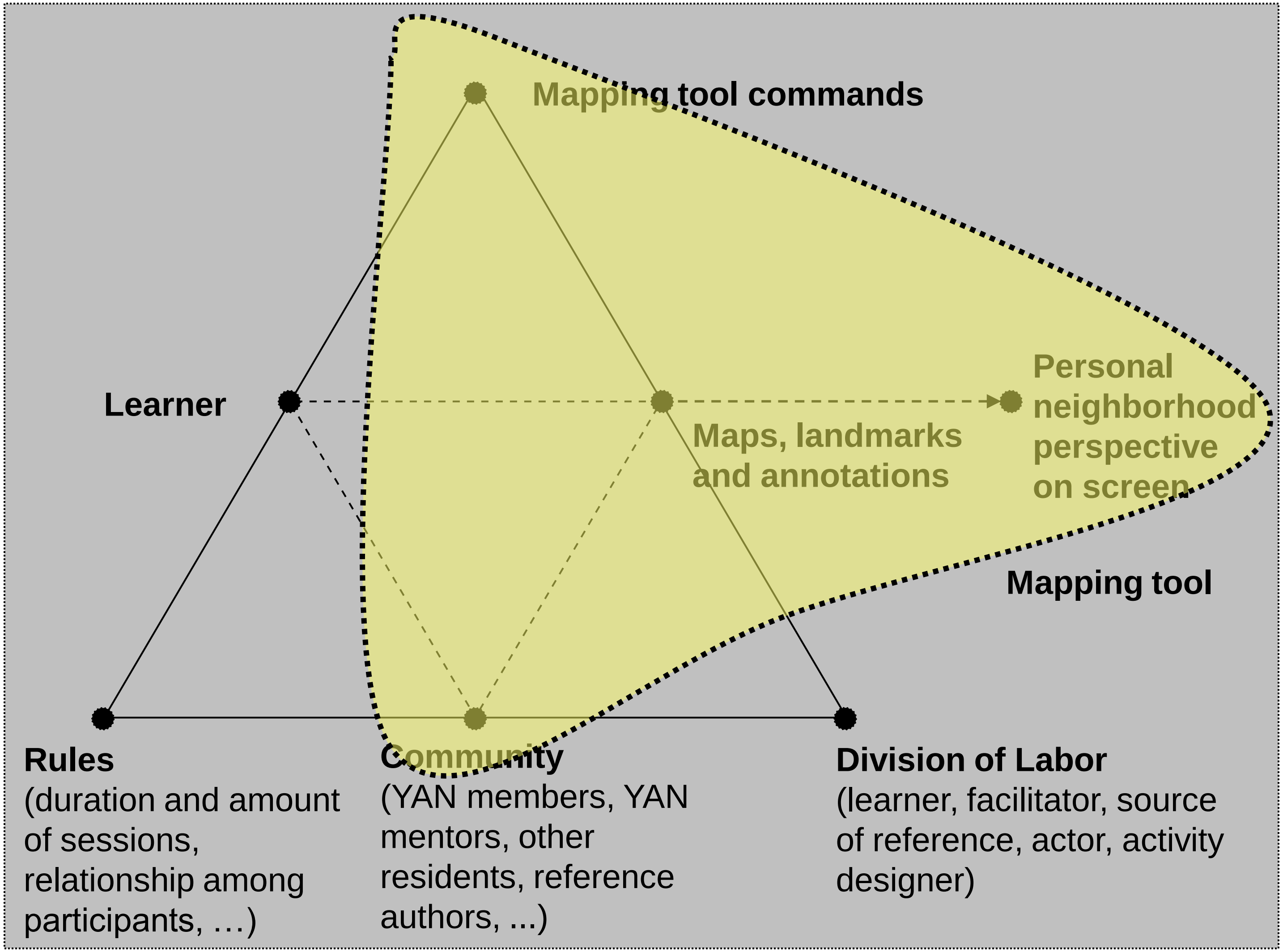
- Use the proposed approach in the creation and analysis of new applications

A neighborhood mapping activity



Mapping tool commands
Learner
Maps, landmarks and annotations
Personal neighborhood perspective on screen
Rules
(duration and amount of sessions, relationship among participants, …)
Community
(YAN members, YAN mentors, other residents, reference authors, ...)
Division of Labor
(learner, facilitator, source of reference, actor, activity designer)

A neighborhood mapping activity


Mapping tool commands
Learner
Maps, landmarks and annotations
Personal neighborhood perspective on screen
Mapping tool
Rules
(duration and amount of sessions, relationship among participants, …)
Community
(YAN members, YAN mentors, other residents, reference authors, ...)
Division of Labor
(learner, facilitator, source of reference, actor, activity designer)

For additional information

- leoburd@media.mit.edu
- http://www.media.mit.edu/~leob/thesis_total.pdf (in Portuguese)