

Development of Software for Educational Activities

Leo Burd

DCA/FEE/UNICAMP
Master's thesis defense

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

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 role of computers within the activity?

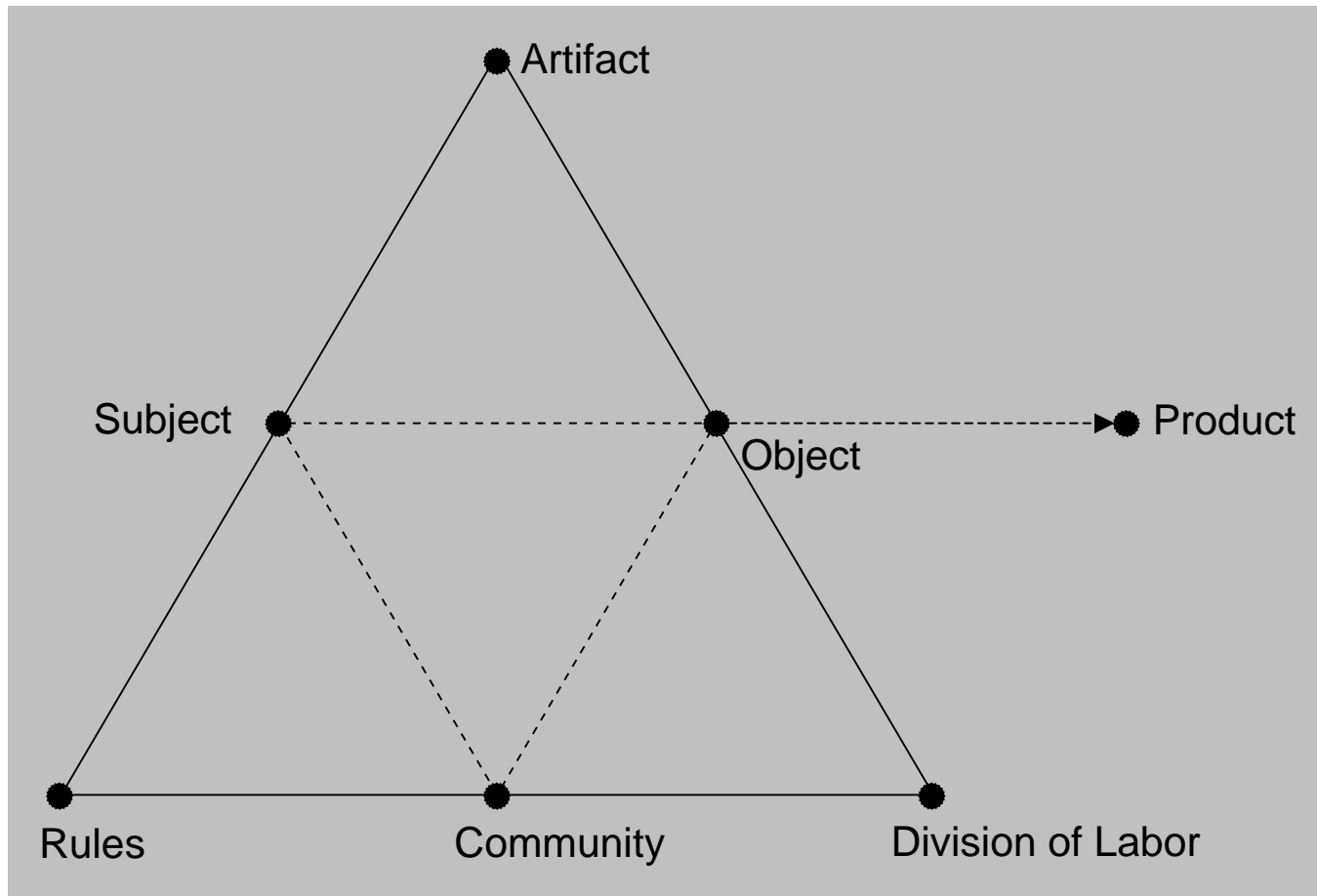
Activity Theory



- 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



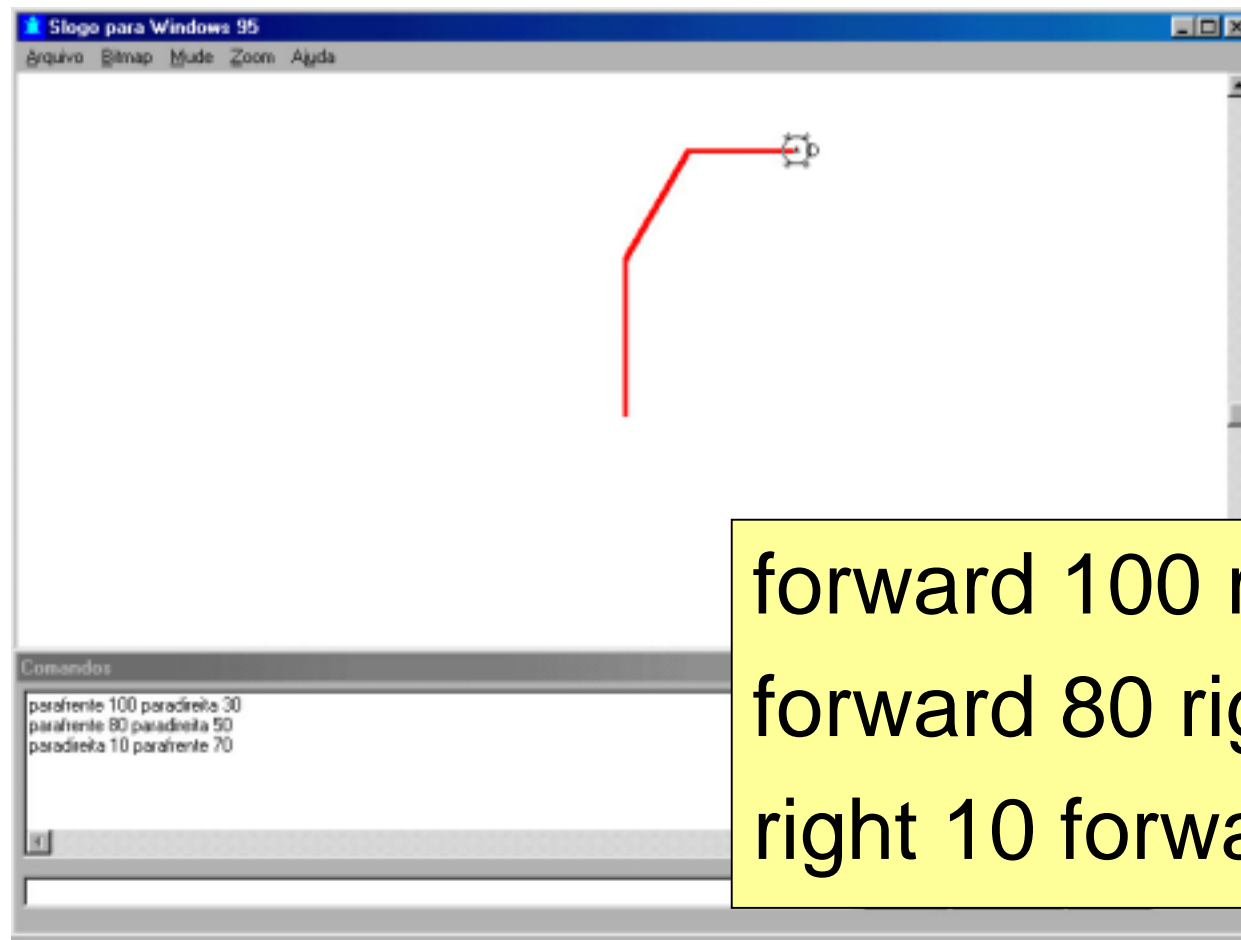
- Hierarchical structure of the activity
- Internalization and externalization
- Object-orientedness
- History and development
- Mediation

Constructionism



- Theory that studies the use of technology on the creation of educational environments
- Created about 20 years ago by Seymour Papert
- Most well-known by Logo, an education software used by millions of people all over the world

Logo's turtle geometry



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?

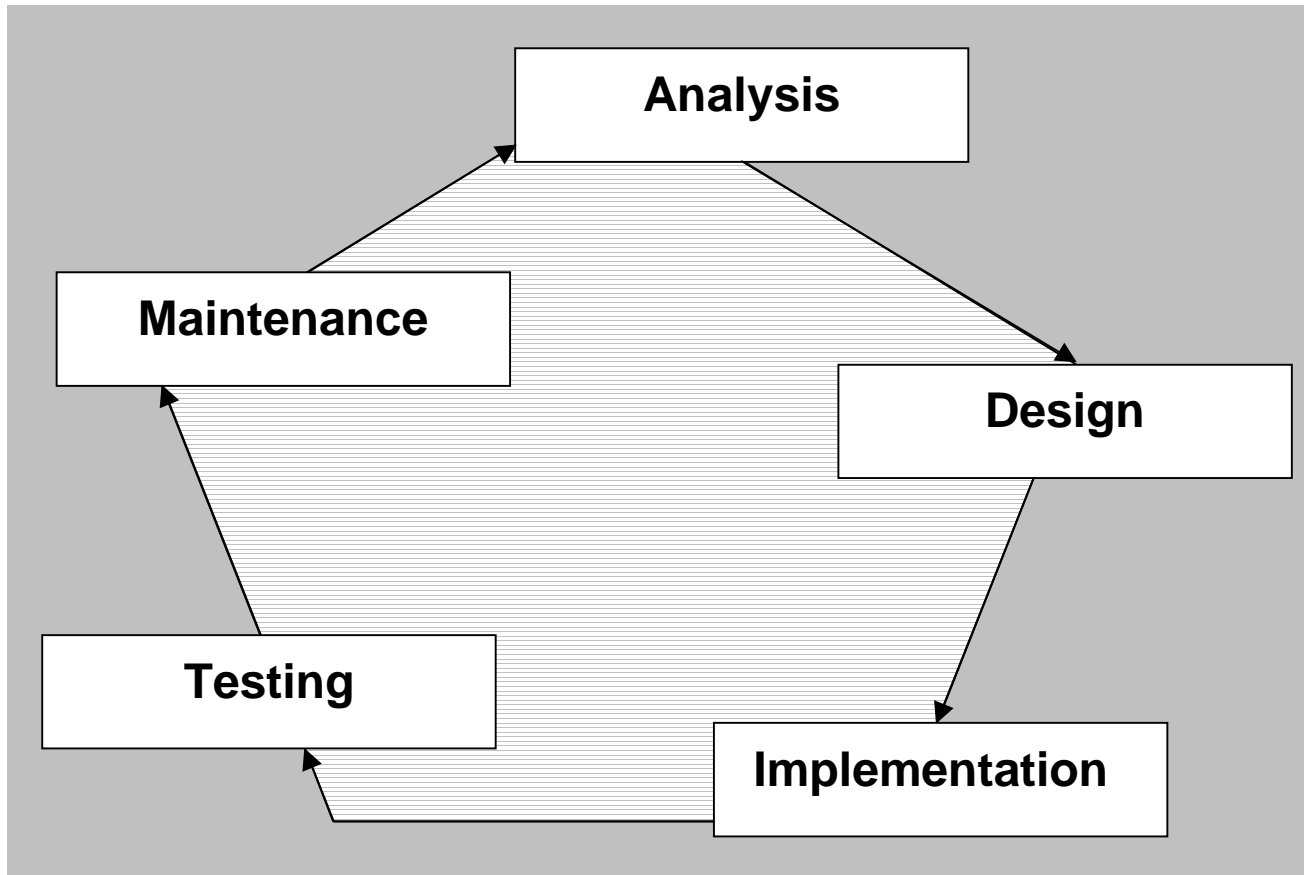
There is a lack of methodologies for the development of constructionist software...

The proposed approach

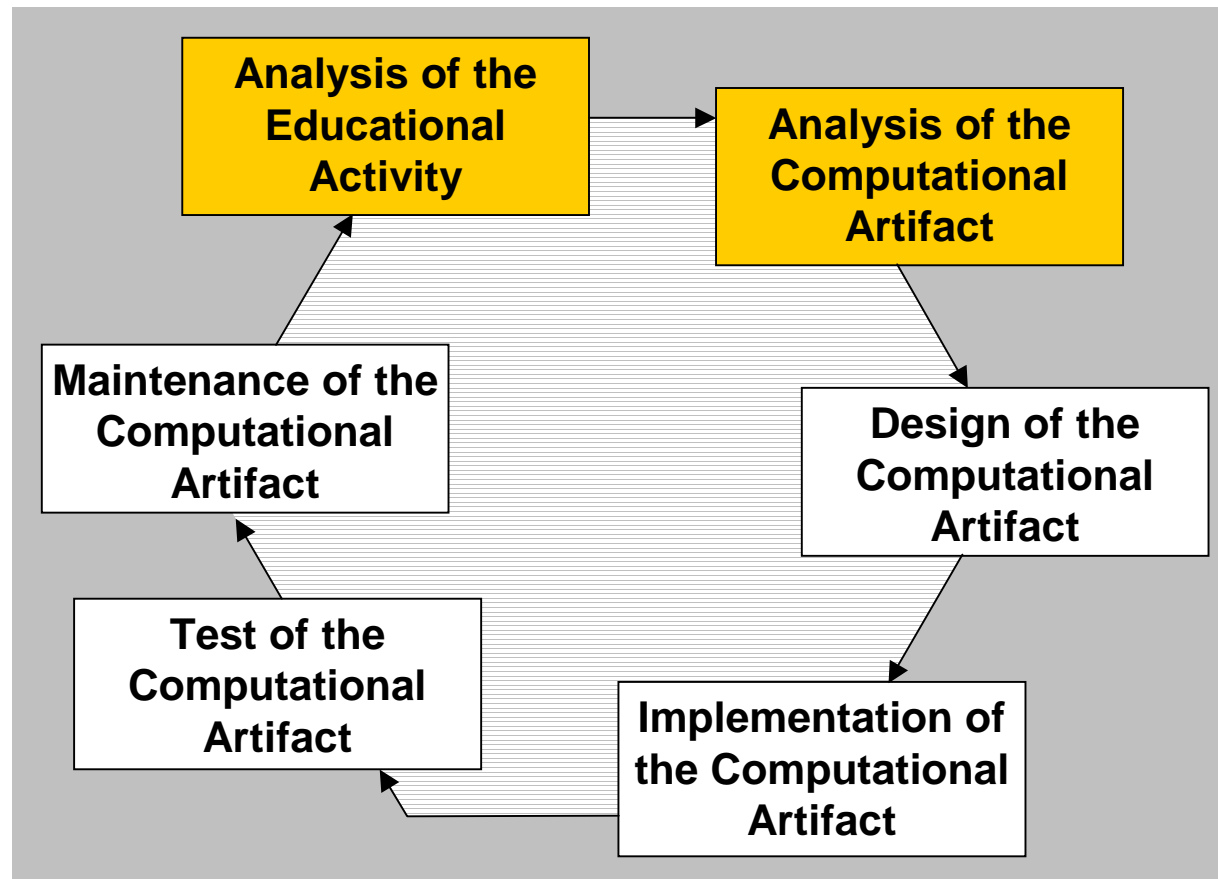


- Is based on a sociocultural and historical analysis of the educational activity
- Identifies the specific contributions of the computer
- Opens space for the other “more technical” software engineering methodologies
- Establishes a permanent communication channel between the user (actor) and the activity designer

The traditional software development life-cycle



The proposed development cycle

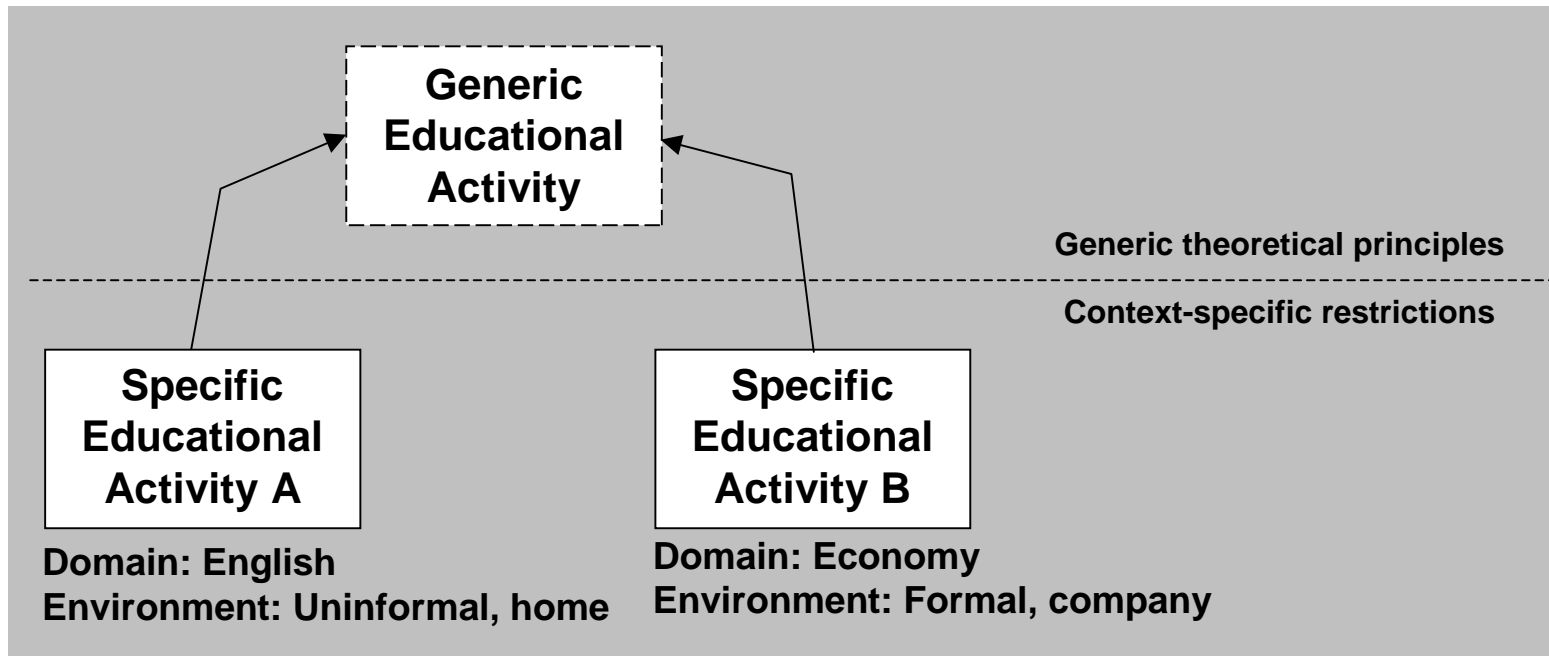


The analysis of the educational activity



- 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

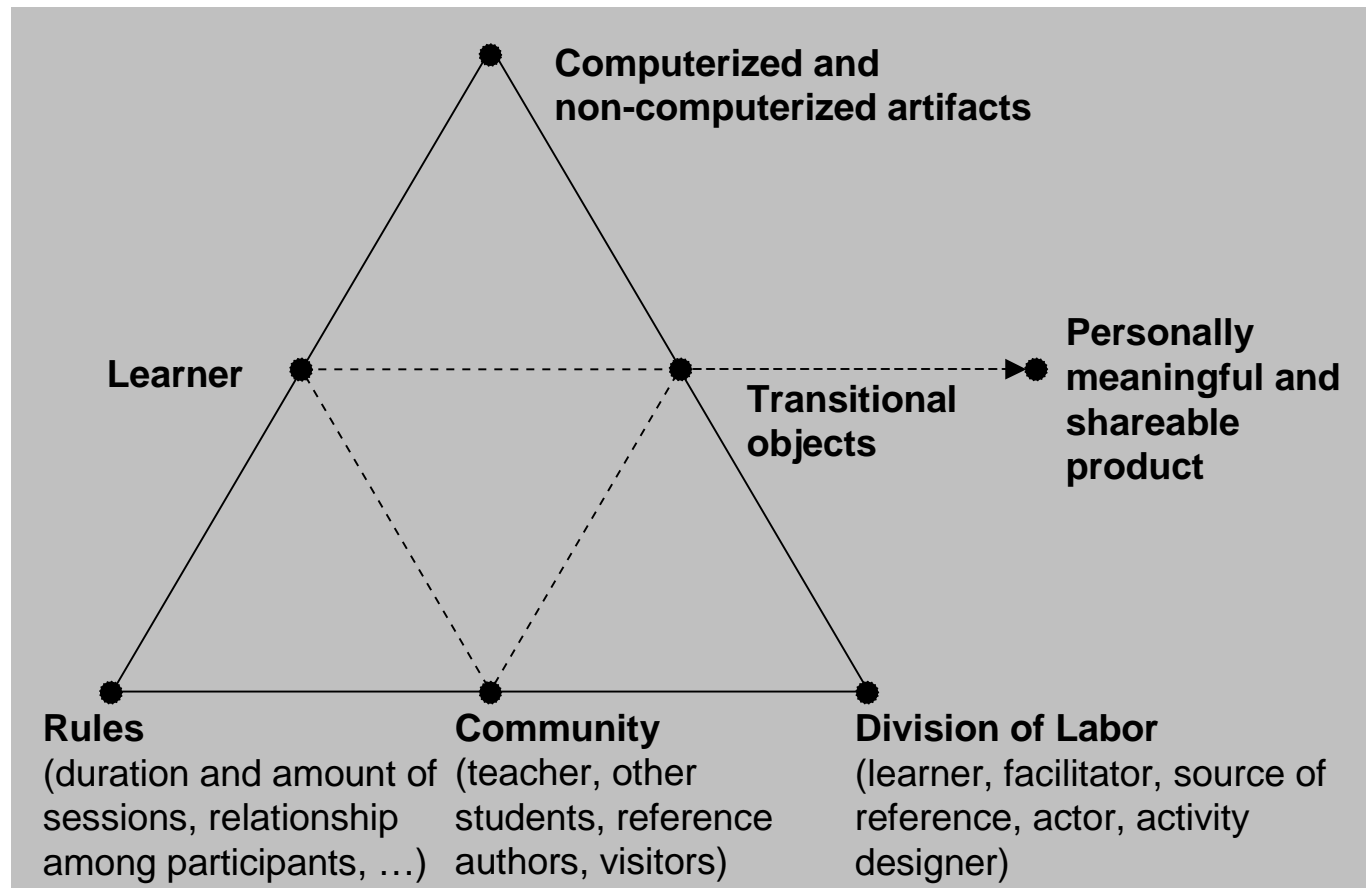


The analysis of the computational artifact

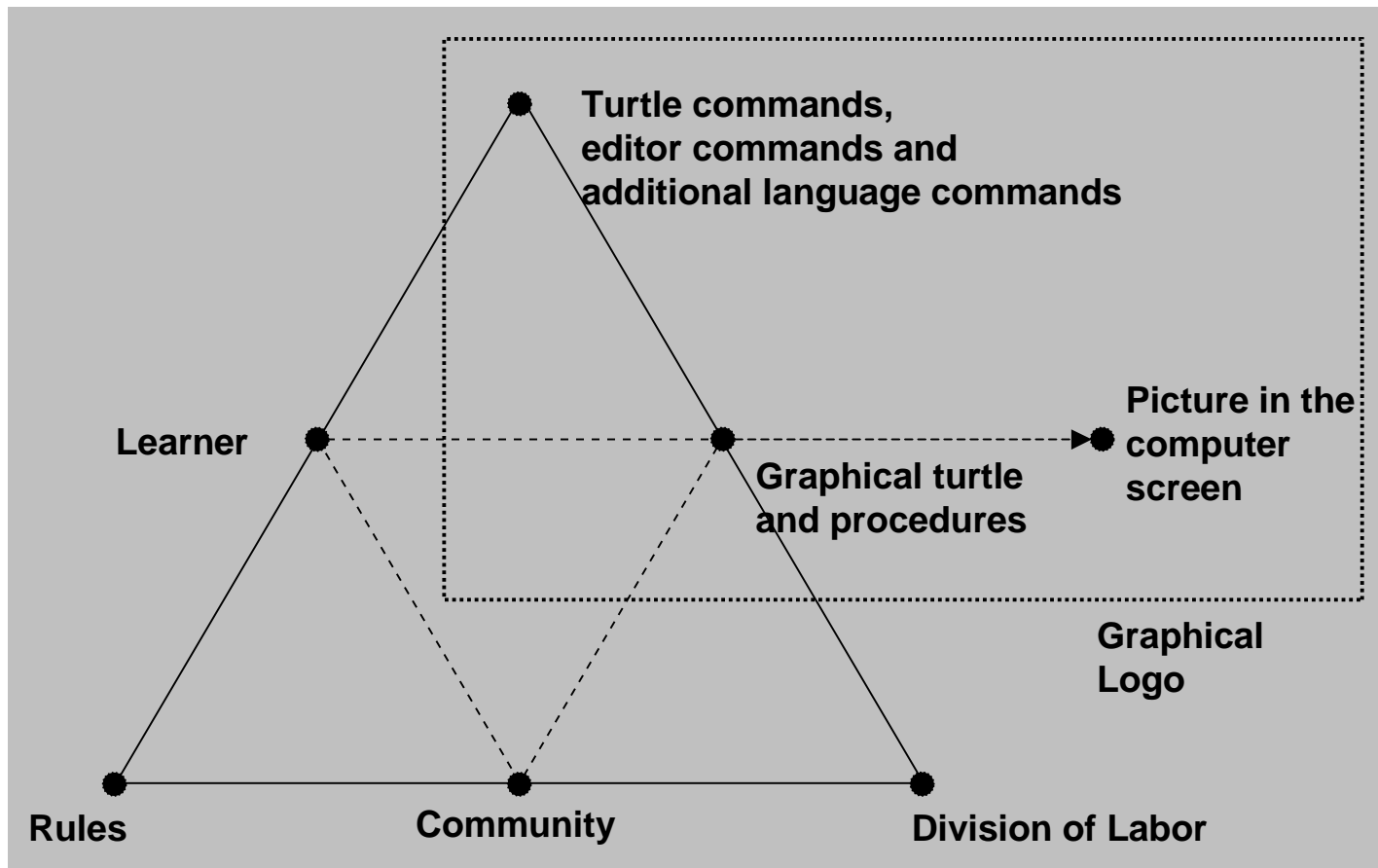


- Describes the role of computers within the activity
- Distinguishes among *computational artifacts*, *digital artifacts*, *digital objects* and *applications*
- Is also driven by guiding questions

A Generic Constructionist Learning Activity



A Graphical Logo Activity

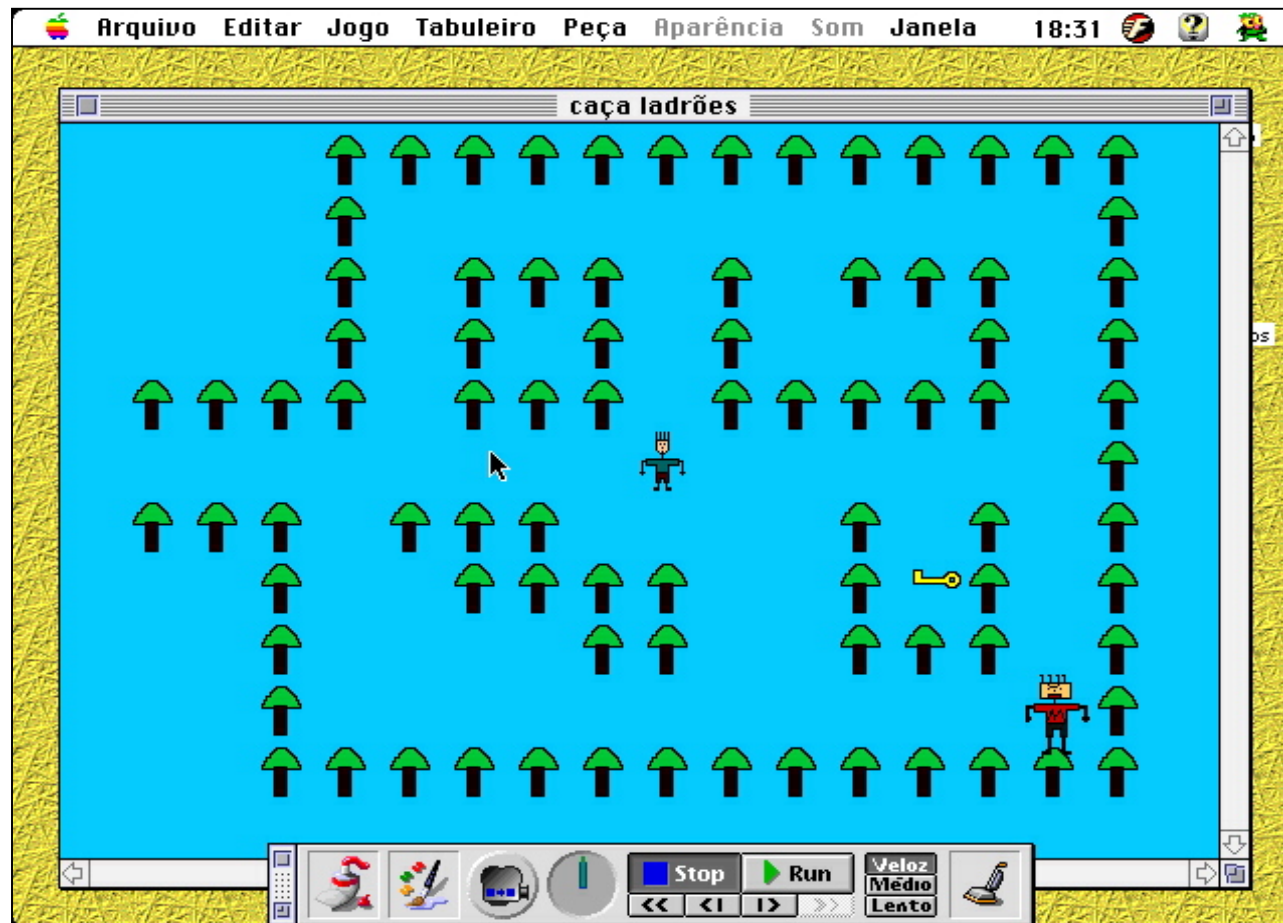


A practical case: The Games Workshop



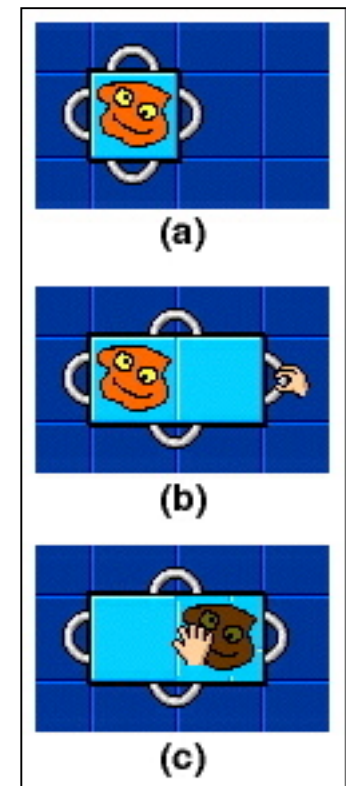
- Institute of the III Millennium - Ibirapuera Park
- “Development of games about the park using the Cocoa software”
- Served as an object to think about this work

A Cocoa screen

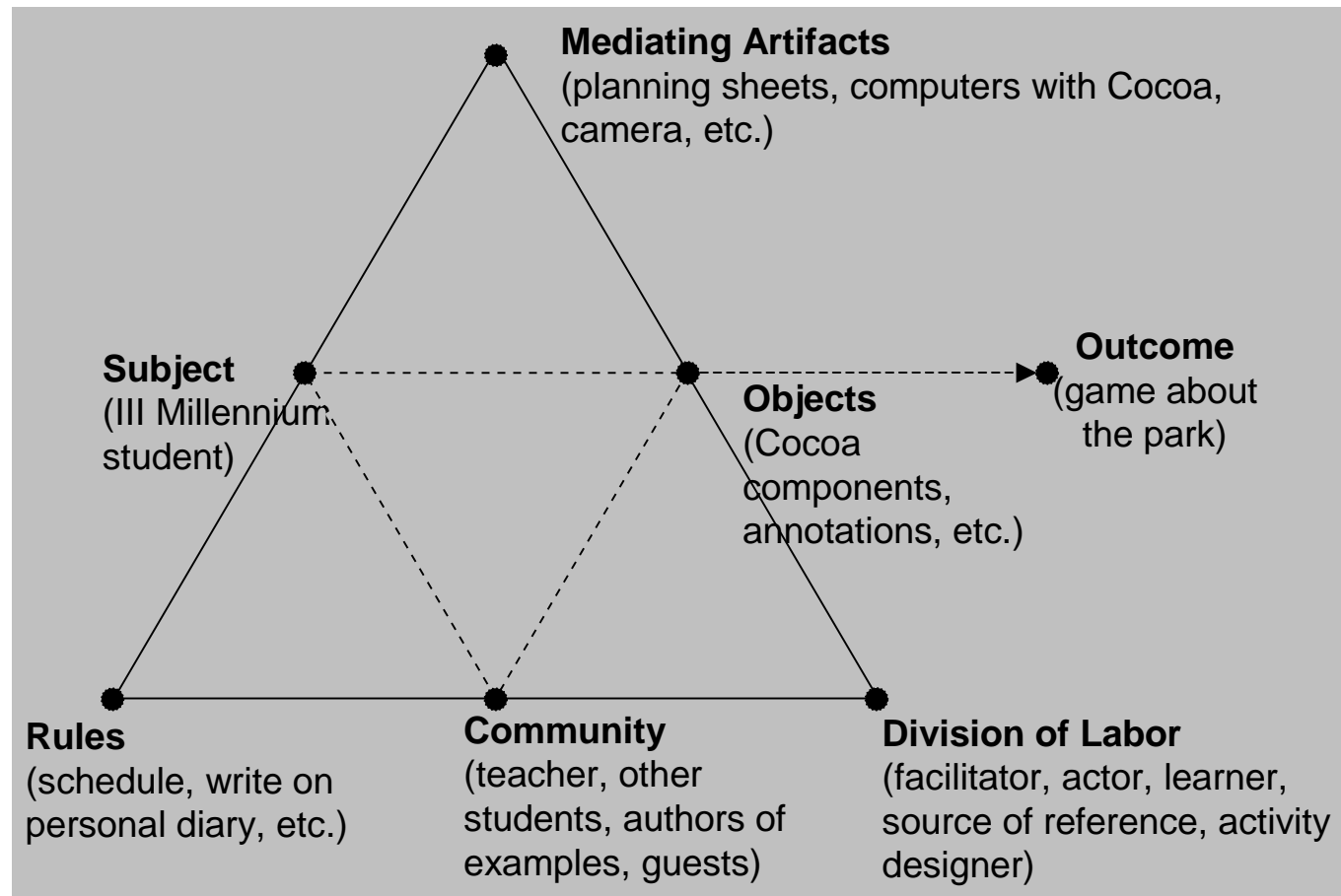


Why Cocoa?

- 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



Conclusions about the Game 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 about the Game 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

Major conclusions of the research



- 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

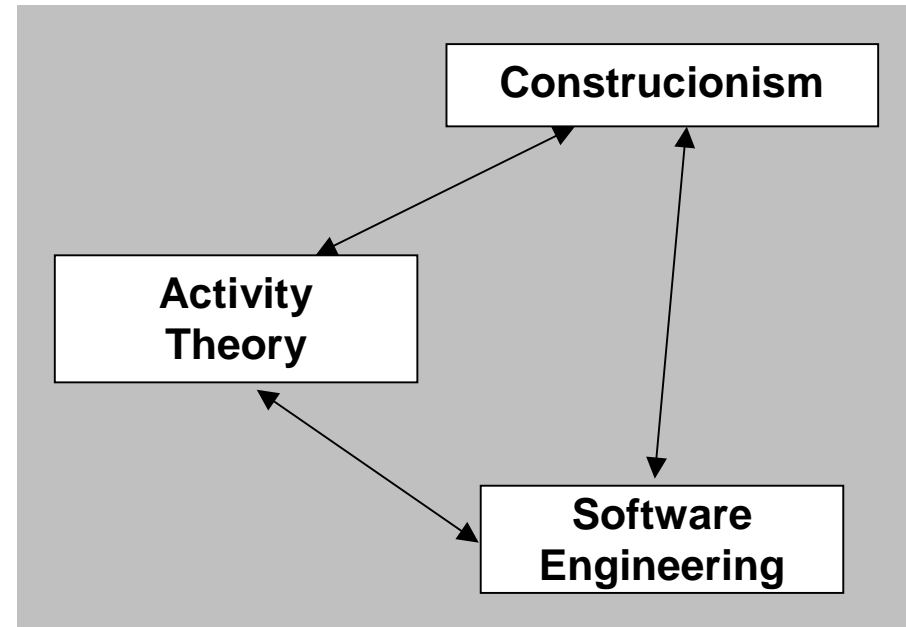
Questions raised about Constructionism



- 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 faced

- 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
- 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 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



- Write articles to spread and improve the ideas
- Use the approach in the creation and analysis of new applications