Building up Development and Design Capabilities in Software through the Creation of Interfaces between Users and Producers: The Case of the Mexican Firms

José Luis Sampedro
Universidad Autónoma Metropolitana, Xochimilco
Área Desarrollo Tecnológico. México, D. F.

Globalics Academy, Lisbon May 30th 2005
1. Aim
2. Research problem and research questions
3. Analytical framework
4. Research strategy
5. Software and technological evolution
6. Structure about the Mexican software sector
7. Stylized facts
1. Aim

The aim of this work is to explain and analyze the process of construction of interfaces between users and producers firms of software and how, through the interfaces, they create knowledge and accumulate technological capabilities.
2. Research problem and research question

- Mexican firms have followed up the production of customer software of the proprietary type.

- However, there are firms that have begun to develop and design customer software of the free type. They have higher possibility to create knowledge and build up technological capabilities in an incremental way.

- How do Mexican firms build up interfaces and why these are important in the creation of knowledge and in the process of technological capability accumulation?

- What kind of interfaces do the Mexican firms create?
3. Analytical framework

Techno-Economic Paradigm (TEP): “A change into the collective conscience that become the common sense of engineers, managers, investors, entrepreneurs…for obtain the maximum efficiency and the best-practice productive” (Freeman and Perez, 1988; Perez, 1986, 2003)

- Structural change
- Technological revolution
- Technological system

- Radical innovation
- Incremental innovation

Invention-Innovation-Diffusion
User-Producer interaction in the process of innovation

(Lundvall, 1985, 1988, 1992)

- Standardization and frequency from the exchange
- Symmetrical and asymmetrical relationships
- Information flows
- Economic and cultural space
- Organizational dimension

Learning by doing

Learning by interactive

Learning by using
Interface and TEP (1)
(Andersen, 1991, 1996)

- Interface can be defined as a coordination between users and producers who exchange different types of information. If a interface is accepted and stable, the information necessary of each agent will be delimited. But, the innovative process will presuppose an information-rich interaction and thereby often presuppose non-standardized interfaces but complex interfaces.
Interface and TEP (2)

Information

- Stability
- Knowledge as part of a system
- Simple and stable interface
- Minimum information flows

Commodity abstraction

- TEP with information flows

Interaction

- delimited
- non-limited

Interactive learning

- Users-Producer interaction
- Greater information flows
- Innovation
- Complex and flexible interfaces
- Idiosyncratic knowledge
- Learning
- Knowledge accumulation
Problems originated in the environment could be solved through the interface.

**Simple interface**
(Commodity abstraction)
- Mature technology
- Difficult to do exchange
- Routines are established under this principle
- Short information flows

**Complex interface**
(Interactive learning)
- New technology
- Flexibility change
- Large information flows
- Learning process
- Knowledge generation
Knowledge as a key input of the interface (1)
(Nonaka y Takeuchi, 1994; Davenport y Prusak 1998; Senker y Faulkner, 1996; Malerba y Orsenigo, 1999)

- **Information**
  - It depends on specific contexts and is relational
  - Flow of messages: data are manipulated for decision-making
  - Judgments and meanings

- **Knowledge**
  - It is created dynamically during the social interaction
  - Beliefs
    - Created starting from messages flows (information)
    - "Knowledge is a fluid combination of experiences, values, contextual information and expert ideas that provide a structure to evaluate and incorporate new experiences and information" (Davenport y Prusak, 1998:5)
### Tacit knowledge

- Personal, it depends from the specific context
- Cognitive dimension (models, diagrams)
- Technical dimension (know how)
- Expertise and practice

### Explicit knowledge

- Articulated
- Transfer through formal and systematic languages (codes)
The learning process and the development of internal capabilities allows firms to improve their productivity and their innovative process at the product, process, and organizational level over time (Maxwell, 1981; Bell, 1984; Bell y Pavitt, 1995; Lall, 1992, 2000; Hobday, 1995, 2000, 2001; Dodgson, 1993).
The basic idea is that capabilities represent abilities to do things, and technological capabilities reflect the mastering of technological activities.

(Dutrénit, 2000; Dutrénit y Vera-Cruz, 2001; Vera-Cruz, 2003)
## Analytical category

<table>
<thead>
<tr>
<th>Concept</th>
<th>Category</th>
</tr>
</thead>
</table>
| **Interface**                 | - Class of interfaces  
                              | - Technological cooperation  
                              | - Interactive process (user-producer linkage) |
| **Knowledge**                 | - Generation of knowledge  
                              | - Relevant knowledge  
                              | - Sources of knowledge  
                              | - Diffusion of knowledge   |
| **Technological capabilities**| - Design of software  
                              | - Developed of software  
                              | - Activities of learning   |
4. Research strategy

Research strategy

Theoretical framework

Research questions

Research strategy

Micro level
Design research

• Research questions
• Theoretical proposal
• Unit of analysis (multiple-units of analysis)
• Logic linking between data and theoretical proposal
• Evaluation of the results
Research questions

1. How do Mexican firms build up interfaces and why these are important in the creation of knowledge and in the process of technological capability accumulation?

   What kind of interfaces do the Mexican firms create?

2. What kind of knowledge (tacit or explicit) is dominant in the process of creation of interfaces and in the develop and design process?

3. What importance does the develop and design process have in the generation of knowledge and in the process of technological capability accumulation?

   Are there differences between proprietary and free context?
Theoretical proposal

The interfaces are a crucial element in the process of generation of knowledge and in the process of technological capability accumulation, where the tacit knowledge is relevant. Unlike the firms that develop and design proprietary software, the firms that develop and design free software have a window of better opportunity to generate new knowledge and to accumulate technological capabilities through the creation of interfaces.
Units of analysis

Free software firms (case)

Interfaces

Develop and design process

Proprietary software firms (case)

Exploratory multiple-case study
<table>
<thead>
<tr>
<th>Sources of information</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Open interview</td>
<td>To leaders of project, and developers of software</td>
</tr>
<tr>
<td>(17 interviewed)</td>
<td></td>
</tr>
<tr>
<td>(2) Direct observation</td>
<td>Imply direct observation about the develop and design of software</td>
</tr>
<tr>
<td>(3) Internal and external documents</td>
<td>Books, magazines, papers, files</td>
</tr>
</tbody>
</table>
5. Software and technological evolution

- Software:
  - Technical: algorithm involves natural and social process
  - Link between the society and technology
  - Root: the code
  - Principal input: knowledge
  - Knowledge that is tacit, indefinite, dynamic…always in evolution

- Customer software: application done in base from the requirements of the users, is specific at sector and firm level
## Evolutive path of the computers, 1950’s-1990’s

<table>
<thead>
<tr>
<th>Type</th>
<th>Events</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mainframes</strong></td>
<td>• 1940’s: rustic software, scientific and military</td>
</tr>
<tr>
<td></td>
<td>• Supply for computer producers</td>
</tr>
<tr>
<td></td>
<td>• 1950’s: industrial using</td>
</tr>
<tr>
<td><strong>Minicomputers</strong></td>
<td>• 1965, industrial introduction in small firms</td>
</tr>
<tr>
<td></td>
<td>• Software supply for computer producers</td>
</tr>
<tr>
<td></td>
<td>• Specialized software</td>
</tr>
<tr>
<td></td>
<td>• IBM separated the prizes of software from hardware</td>
</tr>
<tr>
<td><strong>Microcomputers</strong></td>
<td>• 1970’s: micro processor,</td>
</tr>
<tr>
<td></td>
<td>• Operative systems</td>
</tr>
<tr>
<td></td>
<td>• Industrial applications</td>
</tr>
<tr>
<td><strong>Works stations</strong></td>
<td>• 1981-82</td>
</tr>
<tr>
<td></td>
<td>• CAD, CAE</td>
</tr>
<tr>
<td><strong>PDA</strong></td>
<td>• Personal digital assistant</td>
</tr>
<tr>
<td></td>
<td>• Personal mobile assistant</td>
</tr>
<tr>
<td>1940-1950, software development by computer producers</td>
<td>1964, 360/IBM: software compatible over different computers</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td>Specific software and only for each user</td>
<td>Mass applications</td>
</tr>
<tr>
<td>Cooperation from users into the development for better applications</td>
<td>Software develops by independents firms</td>
</tr>
<tr>
<td>Growth</td>
<td>Consolidation</td>
</tr>
<tr>
<td>-------------------------------------------------------------</td>
<td>----------------------------------------------------</td>
</tr>
<tr>
<td>1970’s: 1500 independent software firms</td>
<td>Dominant design for PC,</td>
</tr>
<tr>
<td>1971: IBM introduces the ‘hard disk’</td>
<td>Down prizes of hardware</td>
</tr>
<tr>
<td>Package software for PC’s</td>
<td>Operative systems and application for PC are homogeneous</td>
</tr>
<tr>
<td>Problems with quality, management, measure, prize</td>
<td></td>
</tr>
<tr>
<td>Networks</td>
<td>Free software</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>1980’s, work in networks:</td>
<td>Richard Stallman, 1980’s</td>
</tr>
<tr>
<td>Intranets e Internet based on PC</td>
<td></td>
</tr>
<tr>
<td>1990’, software determinate by</td>
<td>Software develops into virtual networks</td>
</tr>
<tr>
<td>work in networks, work in teams</td>
<td></td>
</tr>
<tr>
<td>Necessity for development customer software</td>
<td>Open source code</td>
</tr>
</tbody>
</table>
# Rationality

<p>| To improve the quality and the effectiveness from the development |
| Process replication of development |
| Had been possible to manage but not to rationalize |</p>
<table>
<thead>
<tr>
<th>Varieties of Software</th>
<th>Class of software</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Operative Systems</td>
</tr>
<tr>
<td>Package Software</td>
<td>* °</td>
</tr>
<tr>
<td>Specialized Software</td>
<td>° *</td>
</tr>
<tr>
<td>Customer Software</td>
<td>° *</td>
</tr>
</tbody>
</table>

° Free Software, * Proprietary Software
Source: Own elaboration.
6. Structure about the Mexican software sector

- 1992-2001
  - Software: 0.1 / GNP
  - TI: 1% / GNP
  - Software: 7.2% / TI national
  - Hardware: 37% / TI national

1000 – 1500 proprietary software firms

100 free software firms, 1000 developers and 10 000 users
### Proprietory software segment (1)

<table>
<thead>
<tr>
<th>Concept</th>
<th>Employment</th>
<th>Average employment</th>
<th>Firms Num.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Micro</td>
<td>&lt; 15</td>
<td>7</td>
<td>63</td>
</tr>
<tr>
<td>Small</td>
<td>De 16 a 100</td>
<td>60</td>
<td>117</td>
</tr>
<tr>
<td>Medium</td>
<td>De 101 a 250</td>
<td>175</td>
<td>14</td>
</tr>
<tr>
<td>Large</td>
<td>De 251 a 1,000</td>
<td>600</td>
<td>11</td>
</tr>
<tr>
<td>Corporative</td>
<td>&gt; 1,000</td>
<td>1,500</td>
<td>1</td>
</tr>
</tbody>
</table>

Fuente: AMITI, 2001
87% are small and medium size
- < 60 L / firm

6.7% are medium size

5.3 large

One corporative firms, 1500 employees

Competitive firms into the international markets:
- 250 L, and have to grown to 1000
- CCMi (AMITI, 2001)
Free software segment

- 100 enterprises
  - 90% micro-small, < 15 employees
- 1000 developers and 10 000 users
- 2002, 7.9 Free Software / Proprietary Software
- 2004, 9.0 Free Software / Proprietary Software
- Niche: servers
  - 2003: 15 000
  - 2004: 20 000
Mexican software segment

- 2006, sales: $5,000.00 mdd
- 1.5% total annual production
- To create 100,000 local employees
- Demand of IT: services, electronic, government, financial sectors, manufacture
7. Stylized facts

Proprietary context
- Interfaces tend to be simples and standardizes
- Knowledge generated inside of the firms
- Software develops in the context of industrial secret

Free context
- Interfaces tend to be complex, non-standardizes
- Knowledge generated into ‘communities of developers’
- Software develops in the context of ‘free’ and work in teams (networks)
Proprietary software context

IT

Code closed

Traditional model

TEP
Diagram of a interface. Proprietary software
Free software context

IT

TEP

Open source

Communities of developers
Diagram of a interface. Free software
## Toward a taxonomy of interface-knowledge-capabilities

<table>
<thead>
<tr>
<th>Concept</th>
<th>Free software</th>
<th>Proprietary software</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Interface</strong></td>
<td>Knowledge (complex)</td>
<td>Market (stable)</td>
</tr>
<tr>
<td></td>
<td>• Organizational (stable)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Productive (complex) (embedded software)</td>
<td></td>
</tr>
<tr>
<td><strong>Knowledge</strong></td>
<td>Generated into communities of developers</td>
<td>Generated inside of the firm</td>
</tr>
<tr>
<td><strong>Technological capabilities</strong></td>
<td>• Traditional design</td>
<td>• Traditional design</td>
</tr>
<tr>
<td></td>
<td>• Development into the communities</td>
<td>• Traditional development</td>
</tr>
</tbody>
</table>

Source: Own elaboration.
Interfaces: there are types and depths in each context (free and proprietary) that depend on:

- Proprietary context creates traditional and stable interfaces
- Free context creates complex and non-standardized interfaces
- Knowledge: predominant information flows on knowledge flows