

## 3. The Roads Ahead

---

### 3.1 Aplicações Corporativas <sup>249</sup>

It was late March of 2007 and I was in a *kombi*, speeding in the direction of “Centro,” Rio’s commercial district. Taking elevated highways from the campus of Rio’s Federal University on Isla de Fundão, the minivan flew over many of Rio’s favelas, finally landing on Avenida Getúlio Vargas, a block-wide avenue, cleared in mid 20<sup>th</sup> century to modernize the city. I got off at Rua Uruguaiana, a pedestrian street that serves as an entry point to a few remaining blocks of old windy streets, lined with lunch restaurants and office fashion stores, and filled with vendors selling pirated CDs and counterfeit watches. After two blocks, I arrived at Largo da Carioca, a wide square at the heart of Rio’s business district. There I waited for Rodrigo Miranda who was going to take me to “Alta,” a successful Java company was hoping might become one of the sites of my ethnography. Each minute of waiting felt like an hour in Rio’s heat, but I knew it was worth the wait. I had spent the previous few weeks trying unsuccessfully to get myself allowed to come and spend a month inside a Java company. Rodrigo’s introduction could make all the difference.

When Rodrigo arrived a few minutes later, we headed south, crossing Avenida Rio Branco, and entering a tall building that was all too familiar to me. In the previous year I had interviewed people from no less than three companies there. As typical in such office buildings in Rio, the lobby had a system of “optimized” elevators, each going only to a range of ten floors, some of them with long lines. Joining the longest line we ran into several guys whom Rodrigo knew; it turned out all worked for Alta. We followed them to the office, which Rodrigo entered without introducing himself at the door, as if his presence there was perfectly natural. We only paused briefly in the lobby to appreciate the fancy engraved logo on the glass panel that separated the lobby from a large room.

As we entered the large room, I saw three dozen tables, organized into bays and separated by short dividers. Everyone had exactly the same table—including the owners of the company. All but one person looked under thirty. It was like the idea of a Silicon Valley startup from the late 1990s, complete with a bean bag in the corner, perhaps even taken just a little too far. I followed Rodrigo as he shook hands with people, nodding, making our way towards “Felipe” and

---

249 “Enterprise applications” (Portuguese).

“Luis,” seated at desk in the corner. It turned out that they were two of the three co-founders of Alta.

The four of us went to a conference room where Rodrigo introduced me as a Berkeley doctoral student and an *Herculoid*. “Herculoids” was the name of Rodrigo’s private mailing list, which he used mostly to forward technology news. When introducing two subscribers who have not met before, Rodrigo almost always introduced them to each other as “Herculoids.” (“It’s like saying that you are a friend,” he explained to me after we left Alta.)

Rodrigo summarized my research, and talked about what an “opportunity” it had been to get interviewed by me, and how much he recommended that Felipe and Luis agreed to be interviewed as well. The two seemed unsure of what to make of me, but started talking about the company. They were two of the three co-founders, all recent graduates of PUC-Rio’s Computational Engineering program. Felipe was now increasingly doing “the commercial part.” Luis was working on a new company inside Alta. The third co-founder, “Eduardo,” was Alta’s main “technical guy.” Earlier they all used to work with any programming language that a client would ask for, but now they were trying to focus on Java. Ninety nine percent of their work was now in Java, said Felipe. “99.9%,” Luis corrected him.

I start part 3 of this dissertation with a look at Alta, presenting it as the most typical of the contexts that I explored in depth in my fieldwork, providing a background for the later discussion of Lua and Kepler. Calling Alta “typical” would not be entirely accurate, however. Already in 2007, Alta was a highly successful company that in many ways seemed to play all of its cards right. When I returned to Alta in December 2008, I discovered that Alta had grown several times, employing close to a hundred people and occupying three floors of a downtown building. It was now responsible for the main user-facing website of a major Brazilian brand. While Alta was in some way trying to do what many almost all other Rio software companies were trying to do, it was clearly pursuing this strategy more successfully than many of my 2005 interviewees. I will try in this chapter to point out some of the reasons for Alta’s success, but this will not be my focus. Rather, I look at Alta as a successful implementation of a particular approach to peripheral participation in a global occupational world: bringing global technology to local clients. While operating primarily with “standard” foreign technology, this approach involves arms-length relationship with foreign centers of the software world, combined with intense engagement with local organizations. Other chapters in this part of the dissertation introduce two other configurations. One of those involves production of technology with global significance locally, but in relative isolation from local needs. Another, a more complex case, involves an attempt to bring into alignment a wide range of resources, both local and global, in production of a global project aligned with local needs.

## **The Beginnings of Alta**

As I later learned from my interviews with Felipe, Luis and others, the company started in 2003, four years ago before my arrival there, when the three co-founders decided to leave

“Kubix,” another company where they had all worked during most of their university years. Unsatisfied with the limited growth opportunities offered by Kubix, the three decided to start their own company, taking advantage of PUC’s startup incubator, which provided cheap office space on PUC campus, free phone and Internet, as well as help with tasks like marketing.<sup>250</sup> While most software companies in Rio work in software consulting, building custom software for specific clients, entry into the incubator required a business plan that would involve marketing a software *product*. Felipe, Luis and Eduardo wrote a business plan around commercializing Eduardo’s recently completed Master’s thesis, which proposed a method for integrating enterprise information system. Two months later Alta opened its doors on the premises of the incubator with an initial investment of R\$18,000,<sup>251</sup> to which each of the co-founders contributed equally from their Kubix savings.

The product envisioned by Alta’s business plan—“InterJX”—never fully materialized, and from my conversations with Alta’s founders, it is hard to tell exactly how seriously they took it. The conventional wisdom in Rio de Janeiro is that a software *product* company cannot survive in Rio de Janeiro. (“It’s not California,” many developers explain.) The incubator, however, wanted to see a product plan and Eduardo’s thesis made it possible to tell a believable story. The founders themselves seemed to half-believe the plan, putting an intern to work on polishing off Eduardo’s code, while debating among themselves whether to sell licenses for the product or to open source it and hope to make some money on related services. They decided on the latter option, but were not ready to bet on the product’s success, if only for the lack of anything to bet: the starting capital of R\$18,000 hardly provided a financial foundation for a product launch.<sup>252</sup>

When an opportunity came to take a contract with Petrobras, the fledgling entrepreneurs decided to take it.

Felipe: This started right in the beginning, because it was one of Eduardo’s contacts. A friend of his who worked at Petrobras needed a company that could do maintenance on this contract, with intranet in this case, and so he asked if we were available. And we were: “Well, clearly, let’s start billing (*faturar*) and get into Petrobras, which is a large company...” And so we started.<sup>253</sup>

The company proceeded to accept a range of service contracts, which involved software development in different languages, training courses and other tasks.

One of their earliest projects involved Lua. “Rogerio,” a student at PUC who had worked

---

250 For a brief discussion of the Genesis incubator, please see Didier et al. (2005).

251 Between US\$7,000 and US\$9,000.

252 Note that releasing InterJX as open source would imply a strategy half-way between launching a proprietary product (as described in the business plan) and doing services based on third-party software. It would potentially involve less development work and less marketing costs than launching a product and the revenue would primarily come from consulting fees. At the same time, the company would potentially be able to use the product to attract customers—assuming that local customers would have any interest in InterJX over “standard” solutions.

253 See appendix C, “Original Interview Quotes,” (Felipe, April 2007, “Vamo começar a faturar”).

with Felipe and Luis at Kubix, wrote a library for linking Lua with Java—as an assignment for a class—and made a presentation about it at PUC. Rodrigo Miranda attended the presentation and offered to pay for some additional features from Kepler’s grant. As Rodrigo could not pay Rogerio directly, he made a contract with Alta, which then used the money to hire Rogerio to work on the desired features. Luis and Eduardo then joined too, contributing to other Kepler modules as well. As the company was just starting off, any income was welcome. But it was not so much about the money, stressed all the founders. Above all, they wanted to have a good relationship with Rodrigo, who was ten years their senior and knew a lot of people. The investment in personal relations paid off: Rodrigo later matched Alta with its first large project, which the company could then use as a success case when making proposals to other clients. The founders remembered the favor. “I think that if we hadn’t gotten this project Alta wouldn’t even exist today,” said Felipe.<sup>254</sup> (The fact that I was allowed to spend time at Alta and write about it perhaps had a lot to do with this as well.)

As the consulting business grew, InterJX was increasingly put aside.

Felipe: So, from the start we moved our focus a little away from investing in InterJX, investing in integration, and from there it just grew. And as time went by, we were working more and more with development of web applications, development of solutions on demand for clients, right. And InterJX was something we were putting more and more to the side.<sup>255</sup>

When some time later Alta became an official “partner” of a large US-based company offering a Java web development platform, the fate of InterJX was sealed:

Felipe: And a moment came also when we managed to get a partnership with a large, multinational company, which is EIT.<sup>256</sup> We got a partnership with them, and they are one of the main companies that sell integration software. So, we decided to give priority to learning their platform, and to try to offer services on top of it. And this closed the coffin on InterJX. There was a moment when we decided to give priority to working with the technology that is the market leader. Which was EIT. [...] Over there, they have developers, in the United States, Indians, all working to evolve this system. And we here have nobody. [EIT] was light years ahead.<sup>257</sup>

Alta’s founders’ stories about InterJX have a touch of nostalgia, though seemingly not so much the painful nostalgia of squashed dreams as the sentimental memories of lost naiveté. The decisions paid off, however. Early consulting revenues allowed Alta’s founders to hire their most

---

254 See appendix C, “Original Interview Quotes,” (Felipe, April 2007, “Nem existiria”).

255 See appendix C, “Original Interview Quotes,” (Felipe, April 2007, “Deixado cada vez mais de lado”).

256 “EIT” is my pseudonym for Alta’s foreign partner, a large American company based in the San Francisco Bay Area. I refer to their product as “eiWeb.”

257 See appendix C, “Original Interview Quotes,” (Felipe, April 2007, “Fechou a caixa”).

talented friends. The partnership with “EIT,” a household name in Java circles, became a source of larger and larger contracts. Alta’s fortunes were improving steadily and over the years it had attracted many of the former employees of Nas Nuvens. (Rodrigo lamented the loss of good engineers, but saw their departure to Alta as the lesser evil, since Alta allowed some of them to work on Kepler on the side.) Alta was invariably considered a success by those who knew it. When I started frequenting Alta’s office in June 2007, the company’s only problem seemed to be finding place for all the new developers it was hiring (a factor that delayed the starting time of my fieldwork there). This problem got resolved two months later when Alta got a chance to rent an additional, even shinier, office in the same building. Around the same time, Nas Nuvens was asking its employees to come to the office on Saturday to repaint the walls to give the office a slightly more presentable look.

While more successful than some of its competitors, Alta was hardly original in its strategy, using global technology to solve local problems, providing customized IT solutions based on Java web technology to local clients. (As we will see in some of the examples below, though, Alta also to some extent gets to instead adapt local problems to fit the available global technology.) According to the PowerPoint presentation given to the new employees, the company’s focus was

Consulting, integration and development of applications based on  
**new technologies.**

—with the last words shown in big, bold letters. Alta’s founders and employees to whom I talked were equally excited about this company’s commitment to use of the most advanced world-class technology, technology “at the tip.”<sup>258</sup> As later slides explained, in 2007 this meant building web applications in Java, using J2EE. Alta was “A 100% Java company,” declared yet another slide. In reality, of course, conservative clients and the need to maintain legacy systems often required Alta’s engineers to work on somewhat outdated technology. What the company advertised, however, was its *ability* and *willingness* to use the latest solutions. Alta’s developers, or, rather “technology professionals,” as they were called in the same presentation, followed the latest technology news, often in English, and peppered their speech with portuguesified English phrases, from technical terms, to business terms (“*essa delivery*,” “*o deploy*”), or even general phrases (“*é feeling mesmo*”).

The company’s mission statement, stated in the same PowerPoint presentation, however, presented Alta’s other, *local* side:

to transform your desires into reality  
through IT services that match exactly your needs.

---

258 Two Portuguese words are commonly used to describe such technology. One is *tecnologia padrão*. While *padrão* literally means “standard,” it could perhaps be better translated as “world standard,” as it typically connotes the quality rarely found in Brazil, rather than mediocrity that is sometimes suggested by the word “standard” when used in the US. The second term, *tecnologia de ponta*, literally “technology at the tip,” corresponds more easily to the English “bleeding edge.”

Rather than selling a uniform product, Alta cultivated durable relationships with local clients, most of which were quite literally within walking distance from Alta's office. Understanding the clients and their needs was a skill in which Alta's more senior personnel took pride. Alta's relationship with its some of its clients was in fact often so strong and durable, that the company almost acted as a part the IT departments of some of its clients. "Intermercado," a large Brazilian retail conglomerate and Alta's largest client, retained nearly half of Alta's employees on *per month* basis, paying a monthly rate regardless of whether there is any work. Such retained employees thus experienced the typical work cycle of in-house developers: periods of intense round-the-clock work when a project is about to be delivered, followed by quiet time when the details of the next project were being worked out.

If anything distinguished Alta from its competitors clearly, it was the founders' commitment to attracting good developers (mostly from among PUC students) and retaining them with generous compensation. Perhaps taught by their own departure from Kubix, the founders offered free shares in the company for some of the key personnel and paid the salary of others out of their own pockets when necessary. "We had an opportunity to hire Fabio," says one of the co-founders, "but didn't have any money. But he was good, and we couldn't miss this opportunity, so we paid him out of our own pockets." When later talking about Nas Nuvens, he comments: "I knew they weren't doing well when they let Zé go. Zé is really good, you can't let people like him go. They must have been *really* short on money if they couldn't offer him a raise.") A similar, if less radical, approach is extended to all other employees. While the company employs all of its developers as contractors to avoid the complications of Brazilian labor legislation (a nearly universal practice among software companies), it makes a policy of offering them all the benefits that would be due to them had they been hired full time.

## Skol with Fabio

A few weeks after my first visit to Alta, I went back to its office to meet "Fabio," one of Alta's young managers whom I had met briefly on my first visit. Fabio came down and met me outside as soon as I called him, apologizing for being a minute late. (*A professional indeed*, I thought to myself.) He was dressed informally, much like a typical engineer, though with more attention to style. I noticed this mostly because of the stark contrast with the stylish dress shirt that I saw on him the previous time. As I later learned, wearing clothes that strike a careful balance between dressy and hip was the rule for Fabio, and today's t-shirt was an exception. Perhaps noticing me looking at his t-shirt, Fabio explained that he had a busy day, because they had to deliver a project to a client. He would in fact have to go back later to finish some work, he explained.

Fabio asked me if I wanted to sit down right there and get a beer. The space in front of the building was covered with yellow plastic chairs, all decorated with the logo of Skol, a popular beer brand. The number of those chairs, almost all occupied, made me wonder if all of downtown's employees have decided to come out for a beer at the same time. I soon realized that this was precisely the case: it was a Thursday, but Friday was a holiday. "Happy hour," said Fabio

in English. We sat down; Fabio leaned back in the plastic chair, lit a cigarette and asked a waiter for a large bottle of Skol, which got brought to us with two plastic cups.

The conversation turned back to his dress. He normally dresses up, explained Fabio, because he is a minority partner in the firm and ends up interacting with clients a bit. They need to maintain a good impression. Also, he was in mid twenties, and a few years earlier he was even younger. The clients tend to dismiss younger people, he explained. So, he had to compensate for that, to look more serious. “More like an adult,” he said. Otherwise, he explained, they would think: “What does he know?” Also, he sat in the management section of the room, Fabio continued. They are all together in one room, but there is a management corner. They talk to clients in the conference room, but the clients often want to see the office. *Now, imagine*, Fabio said, *that the client walks into the office and sees a guy in torn up clothes, with huge hair, a giant beard.* “What does the client think? He thinks, *Awesome, this guy is a nerd, he probably knows how to program!*” But when they look at the management corner, they want to see people who are well dressed, people who look like they would understand their business.

Fabio came to Alta as an intern in 2003. Before that he worked together with the founders at Kubix. As is typical of “interns” in Rio software companies, Fabio worked full time for most of his university years, though with some flexibility in hours, so that he could schedule his work around classes and homework. (Though, he only really needed to study right before the exams, he said.) Fabio earned a little under R\$400 (US\$200) a month when he started as an intern but saw his salary rise dramatically as he learned more. His experience was typical: interns may earn even less than that (the intern status exempts them from minimum wage requirements, see chapter 2.1), while experienced programmers, even without a completed college degree, often ask for R\$2000-3000 or more. There are limits on engineering salaries, however, say most developers, and to increase their income further the most talented engineers typically move to management. Fabio was starting this transition this year. About a year and a half ago, just before he graduated from the university, Fabio had become a minority partner at Alta.

Some time later, we started talking about Lua. Despite Alta’s claims to be “a 100% Java company,” a number of Alta’s engineers had contributed to Kepler and two were involved with it at the time. Fabio was one of the two. He had started working with Lua back at PUC, Fabio explained. He took a class from Roberto Ierusalimschy, one of Lua’s authors, and another one from Roberto’s spouse. Both had programming assignments in Lua, and in case of Roberto’s class the task was to contribute something of value to the Lua community. While some other PUC graduates that I interviewed resented being required to use a “home-grown” programming language (see chapter 3.3), Fabio took the assignments more positively, since some of his friends at Kubix, including Luis, one of Alta’s co-founders, were fans of Lua. (Luis in turn attributes his interest in using Lua to seeing it used extensively in Tecgraf, a research lab at PUC.) When Fabio later needed a final project for his engineering degree, Rodrigo Miranda, whom he met through Luis, suggested writing a new version of a Lua programming editor, originally written by Luis and based on Eclipse, a popular Java platform. Fabio wrote it together with Rogerio, a close friend who was now also working for Alta.

After finishing college Fabio continued working on the tool, getting paid a small amount from one of Rodrigo’s grants. (As Rogerio said later, the money paid by Rodrigo was hardly

much, but it helped establish a commitment.) It was hard to find time for it, Fabio said, but he wanted to continue participating. He gave me two reasons. When you work with something all the time you get tired of it, he said. He was working with the EIT's "eiWeb" platform all day, which he also found to be "very commercial." Lua work was a diversion. It was not serious commercial technology, not something he would suggest to a client, but it was fun (*divertido*). The idea of Lua as diversion from more "serious" technology reminded me of a slightly less sympathetic term I often heard to describe it—"bonitinho" ("cute"). In this case, however, Fabio seemed to use the word positively.

It was also about being a part of the Lua community, he said then. The word community suggested to me the idea of local ties. I immediately remembered Felipe's story about Alta's involvement with Rodrigo's project. Perhaps he was referring to the fact that he worked on his projects together with his friends at Alta, I thought. (As Rogerio told me in a later interview, working on the project was one of the many things he and Fabio did together on the weekends.) In this case, however, it turned out that Fabio had a different community in mind: Lua's mailing list. When he first started using Lua, he explained, he started reading the Lua list and seeing the announcements of people releasing their code. This helped him realize the meaning of "open source." "It's an exchange community," explained Fabio, hence the need to contribute. But it was also a matter of personal satisfaction. In any community, said Fabio, be it your samba group or your church, you want to be known, be a member ("*integrante da comunidade*"), be some who has done something for the community. From this, you get satisfaction. When he released his editor, people replied, they said "Great," some guy wrote back and helped correct a mistake in Fabio's English used in the code, so there was feedback, and people were contributing back.<sup>259</sup>

"Ok," I said, "I understand the attractiveness of an open source community, but why Lua?" Fabio's answer surprised me. The Lua community offers him a space to participate, he explained. Not so with Java. Consider Java, he continues. There are Java User Groups ("JUGs"), like RioJUG. RioJUG has about a thousand people on the list, but the level of discussion is very low. There are probably three or four people who actually understand the technology and who have good questions. The problem with Rio's Java list, continued Fabio, is that the number of people who understand Java is low, but the number of people who *think* they understand it is high. Why? Because Java is fashionable. The market seeks people who know Java. So the guy gets a book, copies an example, runs it—now he "knows Java." Statistically, continued Fabio, there are may be more "Java programmers" in Brazil than in the US. But the difference is in quality. "Those people here, they don't even know how to use Google!" he exclaimed. "You have to use Google. Google is *pai de todos* ['everyone's daddy']. Before asking a question you have to think: someone probably had this problem before, so go look in Google!" That's something people on the Lua list know how to do. And yes, of course you need to know English to make use of Google, otherwise those thousands of results would be of little use. And half of RioJUG

---

259 As many others, Fabio stresses the contributions by the Lua's *international community*. While the foreign members of the list contribute most of the discussion (the value of which is not to be underestimated), the majority of code is actually contributed by people in Rio de Janeiro who are almost always associated with PUC. The latter fact is routinely downplayed by the Brazilian members in their attempt to present Lua as an *international community*.

people do not even know English. Fabio was on a roll, and his rant continued with increasing passion, moving on to Alta's difficulties in finding good Java programmers and their need to hire "raw guys" (*cara cru*) and teach them from scratch.

Neither Fabio's frustration with the local Java community, nor his fascination for the Lua community are unique to him. The complaint that most Brazilian "programmers" do not know what they are doing is something one can hear every day, both from the graduates of elite programs like PUC's Department of Informatics, and from those who themselves might be the source of frustration for PUC graduates. This un-ending source of complains links to the more general frustration that the Brazilian elite feels with its *povo*. The Portuguese word *povo* is often translated into English as "the people," but this translation cannot capture the separation typically implied between the *povo* and speaker when the word is used by middle class Brazilians. While English "the people" readily brings to mind the opening line of the US Constitution, "We, the people," the word *povo* inevitably connotes the undereducated Brazilian masses with which the Brazil's elite is "stuck," and whose crudeness makes it hard for middle class Brazilians to present a respectable image to the rest of the world.<sup>260</sup>

Fabio's praise for the Lua community is also repeated by everyone who is familiar with the group. On the Brazilian side, most list members are former graduates of PUC-Rio Department of Informatics—Brazil's best computer science department. Even they, however, are eclipsed by the list's foreign members who often can quickly answer the most complicated software questions. The caliber of the foreign subscribers can probably be best explained through self-selection: in order to discover Lua, still a somewhat obscure language today and virtually unknown until a few years ago, most of them went through dozens of programming languages, driven by curiosity and a search for perfection. The active Lua users whom I interviewed in California were among the most technically knowledgeable people I had met in California, and my conversations with them were a fascinating tour of the history of programming languages and computing platforms.

For Brazilian engineers like Fabio, Lua can thus be a ticket into a highly exclusive international (or, one could say, *foreign*) community of developers, and an escape from the mediocrity of local groups like RioJUG. Links to PUC and an early start on Lua gives them relatively easy entry into this community. Gaining comparable standing in a different group, e.g., the Linux kernel developers, would be a lot harder. Eric Raymond (1999) talks about the "ergosphere"—the space of work—as one of the key resources in the open source community. According to Raymond, open source communities are gift cultures, where one gains status by offering gifts to the rest of the community. Open source gifts are solutions to technical problems. Much like in academic research, good problems—the ones that would result in valued gifts that are not too costly to produce—are scarce. Ties to Lua's author and Rodrigo gave PUC graduates

---

260 The following popular Brazilian joke expresses this sentiment. As God was creating the world, an angel was watching his work and after some time asked: "Lord, you made different countries, some, like Russia, are very cold, others, like Arabia, are too hot, you made Japan with earthquakes and China with flooding rivers, the Caribbean Sea suffers from hurricanes and North America from tornadoes. Every place has its own disasters, except for Brazil, a country that seems to be perfect in all respects. Why?" To this the Almighty responded: "Just wait till you see the people that I will put in that land." The joke is so well known in Brazil that it is often just shortened to its punch-line: *espera para ver o povinho*.

like Fabio an opportunity to identify good problems within a small but growing community, or jump into an existing project. The results of their work thus enjoyed ample downloads (one of the key metric of success in open source) and recognition within the community.

This success in the open source community, however, is hard to translate into income in the Brazilian market. If Kepler developed more, it could potentially allow them to make a realistic proposal to their clients, said Fabio. He was confident that this will happen eventually. He talked about Rodrigo's recent idea to "open" Kepler and invite more participation from people outside (see chapter 3.4), saying that this would give people more confidence in Kepler. It won't be a "Kepler team" anymore, but "Kepler community" behind the code, explained Fabio, saying both phrases in English. But for now, he contrasted, there was no "case" of Kepler, no large company using it. And Brazilian clients were not into experimenting. When a client asks: "Why use Java?" The answer is "Because giant companies use it." (Fabio rattles off names of a few large US companies.) Compare this with Kepler, said Fabio. "Why use Kepler? Because it's good." This just does not mean much to the clients. They would ask: "Who is Rodrigo Miranda?" I know him, says Fabio, you know him, but they don't. "Who is Márcio? Who is Tiago?" They do not know. They know IBM and SAP.

The choice between using "good" technology vs. the technology wanted by the clients is not limited to Lua. During my time in Alta in 2007, most of company's work was based on "eiWeb," a proprietary platform provided by EIT, Alta's international partner. There was a strong sense among Alta engineers that eiWeb was no longer the best option, or at least not in all cases. The developers' attention was increasingly turned to the many open source alternatives, which offered a number of advantages: they were free (leaving the customer to pay a larger portion of their budget to Alta for customization), they made it easier for the developers to fix the bugs (because the code was open and there was more free documentation online), and because open source was new and *cool*. For those reasons, Alta tried to move its clients to open source solutions when possible. With the largest clients, however, Alta did not have this luxury. Such clients typically had a prior relationship with EIT, which sold them eiWeb licenses and then offered to recommend a "solution provider"—a local software company that would customize eiWeb for the customer's needs. Alta was one of such providers, not the only one. When a client such as "Intermercado" called Alta saying that EIT suggested them as a company that could implement an eiWeb-based system, Alta's managers had to keep to themselves their opinions about advantages and disadvantages of eiWeb. EIT then typically assigned an software architect to supervise Alta's work and to make sure that Alta was not introducing any open source solutions that would serve as alternatives to upgrading eiWeb to a newer version, at a charge. As the client contemplated whether the expensive upgrade was worth it, Alta's engineers worked with outdated technology.

Some weeks later, when I had already started visiting Alta's office daily and looking at Alta's code, I found myself wondering what version of Java Alta was using. I had not worked with Java for four years myself, but knew that Java 5 came out soon after I started my doctoral program four years earlier and that Java 6 came out in early 2006. I walked over to Fabio's desk to ask him about this. "Java FOUR," he responded emphatically, making sure to enunciate the word "four" very carefully and to load the phrase with contempt to the brim. "Why?" I asked. A guy sitting diagonally from Fabio, who I later learned was Alta's third co-founder Eduardo,

swiveled in his chair, turned around and jumped into the conversation: “This is a very good question. A question we ask ourselves all the time.” But the answer was simple, he explained: Intermercado was using eiWeb 8.2, which used Java 4. They had been talking about upgrading, but this would take a while. Brazilian companies have a tendency to follow the rule of “n-1,” he continued. They always use next-to-last version. When they were deciding on the version of eiWeb, eiWeb 9 was in Beta. Obviously they would not use beta. So, they used eiWeb 8. At some point they will upgrade. Until then Alta was stuck working with Java 4, something it would rather not talk about in the presentation to the new employees.

Inability to use existing solutions meant having to invent workarounds. One of Fabio’s projects required a method for scheduling events. Fabio knew of a good open source solution called Quartz, but its use would not be permitted by EIT’s architect. A scheduling mechanism had already been implemented in eiWeb 10, and the “proper” solution involved paying for an upgrade to the new version. Fabio thus had to find a way to implement his own scheduling method by hand—a job in which he seemed to find little joy. He did not expect his solution to be as good as Quartz. The work therefore involved pointless duplication of effort, a clear violation of cultural norms of software development.

## **Building an Online Store**

A few weeks later I met with Fabio again, at a sushi restaurant near Alta’s office (other Alta developers later described it as “very expensive”). We talked more about Alta, Java, Lua and open source—and the possibility of me spending some time inside Alta. It turned out that Alta’s managers had agreed to let me spend time there (being an *Herculoid* had its benefits), and Fabio was to be my contact person. One problem, said Fabio: there was no place for me to sit. The lack of space got resolved a few weeks later, however, and I finally started spending time at Alta’s office.

It was an early afternoon, and I headed to Alta’s kitchenette to get some coffee. I ran there into Mauricio, one of the developers working with Fabio. “Follow me,” he said and led me to the corridor, where Fabio was smoking with a cup of coffee. We exchanged some small talk, I told an anecdote about my trip to Finland. “Now let’s talk seriously,” said Fabio then. Intermercado people got back to him about the scheduling program that Alta just delivered. They said there needs to be a way to set events that start on one day and go to the other. He already thought about it and it wasn’t so bad. It would require some changes to a database, some to the interface. He started explaining a solution. Mauricio interrupted him: “Why not just have the user enter the beginning and the duration of the event?” “Ah, good idea,” Fabio replied. He started thinking out loud, following the suggestion made by Mauricio. This will make it much easier. So, we just let them do this, and then have JavaScript to show the end time. Again, the database will need to be changed, as well as some of the user interface. Fabio listed the specific things that will need to be changed, walking through the steps that the user will have to go through. He then turned to how long the change would take. “Three days?” he asked Mauricio. “A couple of days to write, a few days to test,” Mauricio responded. “Let’s ask for five,” Fabio summed it up, “three to write, two

to test.” Let’s ask for five, he explained, so that we could then agree to four, if they insist. Finishing early is ok. Better than asking for four days and then having to accept three. He asked if Mauricio could start tonight. Mauricio explained that he had an exam. (It was about 5:30 p.m., and Mauricio was doing a college program at night.) Ok, said Fabio, let’s start tomorrow.

As we walked back to the kitchen, I started thinking of asking Fabio if this was something I could help with. I had been at Alta for a week at this point and had found it hard to get to the details of what people do without getting involved in a project. Fabio anticipated my question. “How much memory does your laptop have?” he asked. “1 gigabyte,” I said. “That’s not enough to run eiWeb,” he sighed. We agreed that I would help with the user interface, which would not require running eiWeb. We went to Fabio’s desk and he showed me the application, then sent me the URL and the password. I spent a few hours playing with JavaScript for the new form. The next day, however Fabio informed me that the issue had turned out to be much more complicated, and in fact the requirements for the project were being re-thought altogether. Instead, he said, I could join him on another project that they were just starting: an e-commerce website for a different client.

Early next week I was sitting next to Fabio at his desk, watching him draw a diagram, which represented the relations between the “business objects” of client’s online store, and was to serve as a foundation for a database. (At this point we had agreed that I would later help with the database design.) As Fabio explained, they had done design “by hand” in the past, but he now wanted to try doing it with a proper tool. As I soon realized, Fabio was using not just a diagramming tool but a UML editor—a tool specifically designed for expressing relationships between software objects, which could later be used to generate code and database design.

Fabio was working without too much haste, explaining to me along the way what he was doing. I was surprised that he appeared to be coming up with the various properties of the objects without having to look them up anywhere or even to pause for thinking. I watched as he created a box for “Product” and then typed in the fields that a “Product” is supposed to have. He then created another box, labeling it “SKU”—the English abbreviation for “Stock Keeping Unit” as I later learned. I thought about the “requirements document” that Fabio had sent me the day before, realizing that it did not have nearly enough detail about what the customer wanted. I asked Fabio how he knew what “business objects” the store needs? *Those are the same for all stores!* he laughed. They had been working with online stores for quite some time, he explained. This particular client did not even have anything specific in mind. They had only a vague idea of what a web store would be like. All they had asked for, explained Fabio, was for Alta to build them “an e-commerce website,” and that this website would be no worse than their competitor’s, plus a few extra things. As I realized later, the request for “a few extra things” did not refer to anything specific either—rather, the client just wanted their website to have *something* that their competitor’s site did not have, leaving it to Fabio to figure out what that something could be. Fabio approached the task by looking at their competitor’s site to see what it did and thinking what they would need to do to allow for the same features and a little more. He then had a long meeting with the client, telling them what his team could and could not offer. Fabio continued adding boxes and attributes as he spoke—the task seemed to consume little attention. At one point he paused, to think about what to do with a particular property. “Let’s see what we did for Intermercado,” he said. He opened the source code for Intermercado’s store, looked at it, then

entered the same property name into the new diagram.

Fabio's work on the diagram seemed to require little creativity, but it did require work, much of which went into aligning boxes. I asked him why it was worth doing. "Because I will only do this once," he responded. "Won't it change?" I asked. "It will, but it won't. You know?" As I understood, the design was going to change as the project develops, but the diagrams would not have to be updated. The diagrams are pretty pictures to be turned in to the client, rather than something to really guide the development. They could be used to generate the first version of the code, which would be the starting point for the implementation, but once generated, the code will then evolve on its own.

A few minutes later the Internet connection suddenly went down. The realization that the connection was gone spread quickly through the room and more and more people joined in humorous "Eeeeeee!," which then gave way to laughter and jokes. It turned out that the power would need to be turned on and off for the whole office in order to restore the connection; everyone started shutting down their computers. People started moving around, talking, joking. Many went to the common area, taking seats on the couch and bean bags. We chatted about random things. The system administrator had already flipped the power switches and the Internet was back, but the people were still talking. "The Internet is back up, you know," said Felipe, the founder. His tone sounded half-jokingly apologetic: he seemed to understand that acting as an authority figure and calling on the employees to get back to work would sound funny in the midst of this free-spirited moment that seemed to show Alta's startup culture at its high point. Indeed, the comment just drew laughter. Nobody hurried to get up.

I did not return to Alta next day, having scheduled some interviews about Lua. When I arrived the day after, Fabio told me he had a new idea about what I could do. There was a new open source thing called "Spring Web Flow," he explained, built on top of Spring, a Java framework that seemed to be on everyone's lips at the time. Fabio wanted me to try using Spring Web Flow to implement a shopping cart. I asked him if he had used it before. No, said Fabio. He hadn't. He wanted someone to try it and to build a proof of concept—a demo that showed that this was possible. And he thought that I could take this task.

I was flattered at getting recruited into Alta's research-and-development efforts, but soon realized that Fabio likely picked the task because it was most appropriate for an unreliable worker such as myself. (A few days ago, while discussing my potential involvement with Alta's projects, Fabio told me he would need to know on what days and what hours I would be in the office. I told him frankly about my time constraints and the need to do some more interviews outside Alta, which seemed to leave him uneasy.) The new task was something I could do at my own pace. And if I were to give up on the task without completing it (as I eventually did), this would not impact the schedules for Alta's projects.

I returned to my laptop to try to get started with Spring Web Flow. I soon realized, however, that before I could get any where, I would need to do a lot of basic installation. I went back to Fabio, to ask him what Alta used as an implementation of J2EE. "Use Jetty," Fabio told me, explaining that this was a new implementation that was much faster than the alternatives. I went back to my seat, googled for Jetty and installed it following instructions I found on the web. I then installed Eclipse, a Java development environment that I knew everyone at Alta was using.

I realized that there had to be a way of starting Jetty from Eclipse, I returned to Fabio's desk for further instructions. He told me to install Jetty Launcher from inside Eclipse. Seeing that I looked lost, another developer offered to show me. Here, he said, traversing Eclipse menus: you install it, then you go here, you put your Jetty Path, then click here, then it runs.

I returned to my desk trying to reproduce what I saw. I did not get very far: my Jetty Launcher and my Eclipse did not seem to like each other. As I understood, Jetty Launcher would not work with the most recent version of Jetty. I returned to Fabio several times with questions. At one point, Leonardo jumped in. Yes, he said, Jetty Launcher requires Jetty 5 and it would not work with Jetty 6. I mentioned that I saw a discussion of this on the web and that someone has offered a patch: a set of changes to Jetty Launcher that made it work with Jetty. I was hoping that Leonardo would tell me if this method worked, but his response was disappointing. He had read about the patch but had not tried it; he was still using Jetty 5. He encouraged me to keep trying with Jetty 6, however, and to tell him if I managed to get it working. Someone had to be the first to use it, so it could as well be me.

I returned to my desk, spending more time reading documentation and forum posts, eventually deciding to give up on Jetty 6 and to install the same version as everyone else was using. I could finally start Jetty from Eclipse, but I now needed to build a trivial application. I tried sample applications from Jetty, Spring, and other projects, but none worked. There were too many new applications involved, and it was impossible to tell which was causing problems. As I headed out for the day, I stopped by Fabio's desk to discuss the matter briefly. Leonardo overheard us again. "Use struts-blank," he said.

I returned to my task the next morning. Using struts-blank resolved a round of problems, but introduced me to the next set. I spent more time reading what I could find on the web and approaching the issue from different angles. I was finding lots of relevant documents, though few were useful. Spring documentation assumed too much, without making clear what one needed to run it. Jetty had documentation for version 6, but not for version 5. Jetty Launcher was missing documentation all together. I returned to Fabio with questions several times, at one point asking him if he had actually ever gotten all of those pieces working together. "No," he responded. "The whole point is to get it working."

After another hands-on demonstration, and another failed attempt at replication, I realized that I was using a different version of Eclipse from everyone else and downloaded another version. Everything was almost working—there seemed to be just one file missing. I returned to Fabio: "What Java are you using?" He gave me a funny look. I insisted on him telling me exactly where he downloaded Java from. "I just do 'sudo aptitude install java'" said Fabio. I returned to my laptop, and typed the four words at the Linux command prompt. I suddenly had the right version of Java and my simple project was finally working. I declared my work for the day done and went to the sofa to work on my field notes.<sup>261</sup>

As this episode illustrates, software work requires a peculiar fusion of globalized and localized activities. Much of that work involves interaction with software developed quite far

---

261 This was far from the task Fabio originally proposed (implementing a shopping cart using Spring Web Flow), which was never completed, as I realized that it was going to take a lot more time than I expected and that my (and Fabio's!) time would be better spent if I limited myself to watching others work.

away and documented in bits and pieces around the Internet.<sup>262</sup> The software and the documentation are quite mobile. In theory, anyone with an Internet connection can download and use them. Furthermore, downloading software never used before in the local context and getting it to work by following documentation on the Internet is part and parcel of software development. In my case, getting the components to work together, was “the whole point,” as Fabio pointed out. A software developer who cannot use the Internet to find out how to solve a problem that is new to his colleagues is of little use to a company like Alta. A developer who always approaches others with questions without first making a *bona fide* attempt to find the answers online would be testing the patience of his colleagues. (My position in the company seemed to allow me to get away with a lot more questions than other developers could afford to ask.) On the other hand, the task of getting downloaded software to work to practice benefits dramatically from proximity to people who have worked with it before. A single phrase uttered by a colleague can substitute for hours of Internet search and trial and error, stress many of my interviewees. Software development consequently becomes an intensely local affair.

As the anecdote shows, the power of local advice and the Internet documents both have much to do with the shared context of work. Fabio’s four words “sudo aptitude install java” brought my laptop in sync with all the Alta’s computers, making sure that the steps that had worked for Fabio and Leonardo would work for me as well. This synchronization was possible, however, because Alta’s machines, configured in Brazil, and my laptop, configured in San Francisco, were already running essentially the same software, Ubuntu Linux 6.10. Continuing synchronization of practice was much simplified by the extent to which the context had been synchronized through earlier work, a long process the beginning of which was described in chapter 2.2.

## Being Local

Around 6 p.m., Eduardo started gathering people. They would be having a cake, he explained to me. It was a company tradition: once a month they got a cake and congratulated all the people who had birthdays that month. This time it was just Leonardo, a recent PUC graduate who had been transitioning to management and was the most recent minority partner. Everyone gathered in the conference room. There were two trays of snacks, a cake and several large bottles of coke. Eduardo suggested that we should do introductions for new people. The first of the new people mentioned that he lived in Niteroi—a city across the bay from Rio, from where a large number of Alta’s employees commuted. After that each of the new people was asked to say whether they thought Niteroi was a better city than Rio. When my turn came, I introduced myself but dodged the Rio-Niteroi question. “And what about Niteroi?” several people demanded. I gave

---

262 Ubuntu, Java, Spring, and Jetty were developed by companies headquartered in London, San Jose, San Mateo, and Los Angeles respectively. (Jetty was originally developed by an Australian company, but is now maintained by a California-based company.) Eclipse is a widely distributed effort, based on a software that originated in an IBM lab in North Carolina. It is now coordinated by a foundation with offices in Ottawa and Portland. Jetty Launcher was developed by a programmer in Ottawa.

a vague response, hinting at a preference for Rio. Fabio, a native of Niteroi, aimed a bottle cap at me, as if ready to throw it I were to side with Rio. Amid loud demands for me to take a clear stand on this important issue, I ended my presentation vaguely, unwilling to step on the neighborhood sensibilities of Alta's global IT professionals, skipping the opportunity to act as a foreign judge of this local rivalry. "Vaseline," snickered Fabio as he put down the bottle cap.

Leonardo started cutting the cake. "Who are you going to give the first piece to?" asked several people. There was some suspense. "I will give the first two pieces at the same time," said Leonardo. He cut two pieces. "Those are for my team," he said, giving them to the two developers who worked under his supervision. He then cut a piece for Eduardo, who was slouching in a chair, looking over the team as a patriarch. He owed everything to Eduardo, explained Leonardo, exaggerating the tone and making a joke out of his public acceptance of Eduardo's authority. The next piece went to Felipe, another co-founder. Luis, the third of the original partners was not there, so there was again suspense as to who is getting the next piece. It went to Fabio. The move caused a murmur. Eduardo and Felipe were unambiguously the bosses of the company. Fabio and Leonardo, on the other hand, were both recent minority partners. Leonardo's move thus appeared acknowledge Fabio's status, while also highlighting the difference between minority partners and everyone else. Leonardo laughed as he gave Fabio his piece, then put down the knife: others could cut their pieces by themselves. Startup spirit aside, Alta did have founders, minority partners and general employees. Fabio and Leonardo had to learn to manage their new status *vis-à-vis* others.

## Culture and Ties

During one of my last weeks in Rio in 2007 I met with Rodrigo Miranda at a café in Copacabana. He had agreed to read a paper I had written for the American Sociological Association and to give me some comments. One of his comments concerned my discussion of the developers' ties to the foreign centers of the practice. Rodrigo suggested that "building ties" was perhaps too strong of a phrase. Most people adopted foreign technology and got quite good at it, explained Rodrigo. They learned to "walk the walk." Many also learned to "talk the talk," adopting the foreign software culture. The latter was harder, Rodrigo argued, because you could not read about it in just one place. It is also more important, since once you had it, people could see it in your eyes that you lived technology. But all this has little to do with building actual social ties to foreign communities, he continued. He had been trying to do it with Kepler and finding it extremely difficult. Most people never try. Look at Alta, he said. They adopted the culture, but without the social ties. Their clients are and may always be local.

Rodrigo was right to some extent. Alta was and continues to be an intensely local affair. The company uses foreign technology and much of foreign culture, without much direct contact with the foreign centers. In contrast to Kepler and Fabio's Lua projects, carried out on the side for "fun," Alta's main line of work is local, both in its location (involving little interaction with people outside Brazil), and in its significance. Such local work is often boring and brings the developers limited cultural dividends in the larger world of software practice. In many ways it

comes down to sales engineering: doing what must be done locally to allow a Brazilian company like Intermercado to use the software supplied by EIT.

The local focus of Alta's work, however, is also a source of strength. In addition to making a profit, Alta builds IT solutions that are actually used by many people. After my departure in 2007, the company proceeded to strengthen its relationship with Intermercado, eventually winning the bid to write software that now controls the front page of one the most popular online stores in Brazil. The months that followed were a period of much work and much learning, Fabio told me when I returned in 2008. He and other Alta's engineers had to learn to build a web application that could handle traffic never faced by any of their earlier applications —or by any software based on Kepler, I can add. (Another local contractor of Intermercado, obliged by Intermercado to work together with Fabio's team, was instrumental in this learning.) Local focus was bringing Alta to projects the scale of which made them exciting and a great source of war stories. In the same months, Fabio also finally stopped participating in Lua-related projects. There just was no time for such games.