

3.3 Fast and Patriotic

Fifteen years after its first version was developed, Lua is a fairly popular language, used in a number of well-known software products, both commercial and open source. The work on spillover effects in innovation may lead us to think that Lua's success would present an important opportunity for local economic development: local companies could take advantage of their proximity to PUC-Rio to gain better understanding of the language and its future directions, finding better use for Lua in their products and engaging in related innovation. This is not the case. At the time when I was doing my fieldwork in 2007, Lua was largely unused in Brazil. Apart from Tecgraf, Nas Nuvens and two other small companies incubated at PUC, Roberto Ierusalimschy knew of no Brazilian companies using Lua. If local companies were using Lua, they were not advertising this fact. Roberto remembered only a few occasions when local companies had entered into contact with the Lua team, none of which led to any extended collaboration. In my five months in Rio that year I managed to find just one more company using Lua in Rio de Janeiro, bringing the total to five. By the end of 2008, three of those five companies were either moving away from Lua or had abandoned it altogether. The situation was promising to change due to the inclusion of Lua in the Brazilian standard for digital television (a story related in the end of this chapter), but Lua was yet to gain wide use in Brazil.

There are several reasons for this lack of local adoption. Some of my interviewees point, sometimes with much frustration, to the Lua team's seeming lack of interest in expanding Lua's use in Brazil. In fact, while Lua's authors lamented the lack in local use of Lua in print (Ierusalimschy et al. 2007), I could see few signs of efforts towards helping local adoption. At the time of my fieldwork in 2007, for example, Lua had no Portuguese documentation—an issue that did not seem to cause much concern for Roberto Ierusalimschy. (See chapter 1.2.) A closely related reason is the seeming lack of fit between what Lua offers and what the typical needs of the local industry. Lua provides clear value for two kinds of software projects: desktop software with high performance requirements (for instance games) and small devices that cannot run the more popular programming languages. Both projects involve making *products*. Rio's software industry, however, focuses almost entirely on services to local organizations, which typically involves building web-based systems. Lua offers few obvious advantages in this domain. This lack of fit, however, can be seen as a symptom rather than a cause of the disconnect between Lua and the local industry. Lua's authors gradually adapted the language for the needs of foreign industry largely *because* of their lack of strong ties to the local industry. (In the next chapter we will look at a project that has aimed to make Lua useful for the sort of projects that the local companies do and the difficulties involved in this task.)

Lua's disconnection from the local industry exemplifies a more general pattern of lack of ties between industry and academic research in Brazil. There may be several reasons for such lack of ties. The main proximate reason is the government policy. Brazil's government funds academic research in accordance with the perceived academic success of each department and university. Such success is evaluated quantitatively and involves as an important component a

metric of “intellectual production,” measured by the number of publications. The publications are weighed by the rating that CAPES assigns to each journal. (The weighing system is known as “QUALIS.”) In 2007, the list of accepted journals for computer science included 241 items, rated as A, B, or C based on the number of citations in two citation databases: ISI/JCR and CiteSeer.³⁰¹ Both databases index almost exclusively English publications. Consequently, for computer science, only those Brazilian journals that publish articles in English are given rating above C, and none of them are rated “A” (Qualis 2009). Computer science researchers thus have good reasons to publish abroad. This requires choosing problems that are of interest to their foreign colleagues.

A system of government funding that measures success by local use of research could shift this balance. The policy of giving incentives for publishing in foreign journals and conferences is not without merit, however, and its rationale aligns with the other, distal, reason for the disconnect. Brazilian computer science researchers are located at the periphery of academic computer science. Brazilian software developers working in industry are similarly located at the periphery of their professional world. Both groups must focus on building ties to the centers of their practice, both in order to keep their practice synchronized with the foreign models and in order to act as legitimate representatives of the practice locally. Successful publication in foreign journals is the best way of demonstrating that the research is up to “world standards.” Focusing on the needs of the local industry may lead the researchers away from the central problems of the global practice. Moreover, it may potentially undermine their credibility in the eyes of the *local* industry.

Local firms similarly find it safer to stick with “standard” technology, choosing Sun’s Java or Microsoft’s .Net over PUC-Rio’s Lua. Use of locally produced research is risky since the quality of such work cannot be easily assessed. Even in cases where the developers may believe in the technical superiority of the local product, choosing them may be unwise, as it may scare the clients. (See Fabio’s discussion of Kepler in chapter 3.1.) Companies thus often fall back on the safer assumption that local technology can be ignored as irrelevant.

This creates difficult questions about what Lua could do for the country and the city where it was developed. To understand the challenges of putting Lua to use and of creating the kind of linkages that could help Lua bring about economic development, we would also have to look at the challenges of running a global project from the periphery. As we saw in the previous chapter, Lua’s success was not planned by its authors and in many ways “just happened.” This does not mean, however, that the future success of Lua poses no challenges. In fact, it only makes the situation more complex for the authors, who do not fully understand the factors that led Lua to its current position and hence have a limited idea of what awaits it in future. While such problems may be faced by developers located at the centers of the software world, we will see that some of the challenges have much to do with the peripherality of Lua’s authors.

301 <http://scientific.thomsonreuters.com/training/jcr/> and <http://citeseer.ist.psu.edu/impact.html>

A Little Bit of Actual Patriotism

Even as the use of Lua declined at Tecgraf in the recent years, many of the original users of Lua, such as Antônio and Silvio, have continued to read the Lua list and maintained an interest in the language. Some of them have contributed code libraries to Lua. In addition to saying that they follow Lua because they are currently using it in some projects, Lua users at PUC typically mention two factors contributing to their interest in the language: Lua's origin at PUC, discussed as a matter of sentimental or personal attachment, and the desire to see Lua attain the success that it deserves:

Silvio: I continue to use Lua in Tecgraf projects, so there is a practical reason [for following the list] as things happen in Lua, that's of interest to me. [...] So I want to be a part of anything that happens [on the list] because it may affect me in a project. The other reason is that I love this, I am an enthusiast of this language, I love Lua, I think it's awesome [*um barato*] and I *like* to read the discussions, I actually do it because I *enjoy* it. So there are those two aspects.

When I ask Silvio what he means by “loving” Lua, he explains:

Silvio: Yes, because I've accompanied Lua since the beginning, right? Roberto was my Master's advisor. I mean: undergraduate, masters and Ph.D. I have a friendship tie with him. Every now and then we go and have lunch together, he tells me what's happening, I don't know what. So I very much live in this world of this language and I do enjoy it, being a part, seeing what happens, talking to Roberto, exchanging ideas. So it's like this. It's something I like, really like. Because of the people involved. It's a *fantastic* piece of work.³⁰²

Other members of the local Lua community at PUC often stress the same reasons: Lua is an amazing piece of software demonstrating the genius of Roberto Ierusalimsky, a person who they consider a friend or a mentor; they feel a connection to Lua having seen it from the days when it was completely unknown. Many add that Lua also brings prestige to PUC—a university with which they are themselves affiliated. The fact that Lua was developed in Brazil or in Rio de Janeiro is rarely mentioned.

In early March 2007, as I was reading through archives of Python-Brasil—a Brazilian mailing list dedicated to the Python programming language, I came across a thread entitled “Python—Lua.” The thread started with a request for comments about Lua and a question about its advantages and disadvantages compared to Python. While such questions about “competing” programming languages sometimes invite hostile responses, most of the replies were quite positive. One of them read:

Python has a more elegant syntax, a larger community, a more diversified field of use, better interoperability.

302 See appendix C, “Original Interview Quotes,” (Silvio, May 2007, “O outro motivo”).

Lua is faster, leaner, more patriotic, more adequate for “embedded systems” [in English] and has a VM for Palm [...].³⁰³

The next day, I mentioned the thread to Rodrigo Miranda, saying that “Python people” were discussing Lua on Python-Brasil. *What did they say?* asked Rodrigo. “They said it was fast and patriotic,” I summarized the discussion. “It is fast,” agreed Rodrigo. Noticing that he confirmed only half of the statement, I asked him explicitly: “Is it patriotic?” “I am not into this kind of stuff,” Rodrigo responded.

While stressing their personal connection to Lua’s authors and Lua’s connection to the university, PUC users of Lua only infrequently bring up the fact that Lua was developed in Brazil as a reason for supporting it. When asked about this explicitly, some, like Rodrigo Miranda, explicitly deny any interest in “Brazilian” software, sometimes referring to such “nationalistic” sentiments as demonstrating narrow-mindedness or even lack of education.³⁰⁴ Others admit such feelings after some hesitation:

Yuri: But it’s not a matter of Lua being a Brazilian language, I don’t know...

Antônio: [Pause]. There is a little bit of this too. Because I know people who made it. I wanted to help promote it in some way. [Lists several reasons for promoting Lua.] But there is a little bit of this too, definitely. Of pride, of knowing where it came from, of promoting national software. Definitely.³⁰⁵

Saying that he supports Lua as a case of “national software” clearly makes Antônio uncomfortable. He says this only when asked directly, and only after first making clear that he has many other reasons for supporting the language. As in nearly all other interviews, the answer to my question about Lua as “Brazilian software” comes only after a long pause.

For many educated Brazilians, “patriotic” is a bad word, especially when used in the context of their area of professional expertise. Being “patriotic” or asking what is good for Brazil is hardly “cool.” “Brazil” is a package of problems that one must live with, not something to be excited about. Rodrigo and others often say “we” when talking about the Brazilians, but the solidarity that is connoted by this “we” is the solidarity of prison-mates. When used by engineers in the context of technology, however, the word becomes particularly dangerous, as it may suggest parochialism and inability to grasp the values a larger, global technical culture. A good engineer should not get their judgment swayed by nationalistic feelings, say the developers. For those in their thirties and younger, who were consumers but not producers of technology during the Market Reserve years, Brazil’s technological “backwardness” (*atraso*) in 1970s and 1980s provides ample proof.

303 A message to python-brasil, February 2007, my translation.

304 Though many of the people I have talked to alluded to the association between nationalism and lack of education, the author of “fast and patriotic” comment is currently pursuing a Ph.D. in one of Brazil’s best computer science departments. (The comment of course may be sarcastic.)

305 See appendix C, “Original Interview Quotes,” (Antônio, May 2007, “Um software nacional”).

The perception of parochialism that casts its shadow over any local projects (and which can only be disarmed by personal acquaintance or international success), can be illustrated by a quote from Ricardo, who was introduced to Lua in 1998 as a student at PUC.

Ricardo: I remember that we had to do a [class] project in C and she [Roberto's spouse, also a PUC professor] taught a new language, which just existed for a few years, invented at PUC and called "Lua." I looked at that and was like: "Eew! A language invented here at PUC? How stupid! I am not going to learn this. I'll never use it in my professional life! What will I do with it? And I remember there being two parts to the assignment that she sent us. One part was in C, another part in Lua. And a girl who was doing a part of the assignment with me... [I told her] "Here, do the part in Lua, because I am not going to learn this stuff, I don't want to know about Lua. I'll do the part in C, which is more interesting, since I'll use it."³⁰⁶

Despite the fact that the class assignment introduced Lua in the exact context where it was strongest (as a part of a Lua / C combination), Ricardo avoids any contact with a local language, referring to the very idea of a programming language developed at PUC as "stupid" (and relegating the task to a woman).

Responding to such clear expressions of prejudice is about the only context in which Lua users introduce—with hesitation—the topic of Lua's Brazilian origin.

Silvio: I think it's great that Roberto managed this... There is also this other thing... It's like... I already saw lots of prejudice against this language. It's impressive how this happens. Like this: I saw one of our clients, a person inside a company, say the following: "Oh no, I am here trying to decide whether to use Lua or this Microsoft application. But I think I will go with Microsoft, because if I use Lua and run into difficulties, my boss will think I am crazy. And if I use Microsoft and run into issues, that's not a problem, because this happens every day." It's *ridiculous* to think like this. I feel ashamed to see someone talk about it this way. I mean... Instead of, being able to use... Instead of *incentivizing* work that was done completely in this country... [For] the guy not to incentivize it out of *pure prejudice*... I think it's totally ridiculous... So... Those kind of things motivate me to follow Lua, to try to use it. Because I see that the list, most of the people on the list are not from Brazil, most are foreigners. I saw that Lua has to succeed abroad to gain acceptance at home. [Pause.] In other words, it's a project that I think is fantastic, which I really like. I see Roberto's struggle to make Lua work, I see the work he has to do. Lua reflects, deep down, his genius. And I think Roberto is very good. [Pause.] For those things, I really like the language and have a pleasure to follow its growth. For this reason I don't leave the list, continue reading it. Even if I don't say anything, I stay on the list seeing what

306 See appendix C, "Original Interview Quotes," (Ricardo, July 2007, "Que bobagem!").

happens.³⁰⁷

Silvio mentions in this quote—only in passing—the possibility of “incentivizing” the work done in Brazil. This course of action, however, is presented only in contrast to the prejudice that Lua has faced in Brazil over most of its history.

As I mentioned in the previous chapter, American users often say that they “take a global perspective” on things like Lua, concerning themselves relatively little with where such software comes from, as long as it is presented in fluent English in a way that shows technical competence of the authors. As indicated by the earlier quote from Ricardo and the story related by Silvio, Brazilian software developers often find that they do not have the luxury of such “global perspective.” Often working in contexts where their competence is questioned on a regular basis, they avoid diligently any associations that may bring accusations of parochialism. Unlike Rich, who becomes convinced of Lua’s bright future by reading Roberto Ierusalimschy’s book and noting the demonstrated global competence, potential users of Lua in Brazil often themselves lack the cultural skills needed for deciding whether a programming language book written in English by a Brazilian author demonstrates the command of the global software culture or is a failed attempt to fake it. They thus find it safer to stick with tools the global status of which cannot be questioned.

While Silvio is disappointed by the “prejudice” with which Brazilians approach anything developed in Brazil, he realizes that he has to accept some of the basic principles underlying this prejudice. To argue for Lua as a case of national software would be to invite yet stronger prejudice, feeding the suspicion that those who support Lua do so for reasons of narrow-minded nationalism. The best way to counteract the prejudice toward a home-grown language is to downplay Lua’s Brazilian connections and to look for global credentials that are valued locally. “I saw that Lua has to succeed abroad to gain acceptance at home,” says Silvio.

Personal ties beat prejudice. A year later after dismissing Lua as “stupid” in college, Ricardo joined Nas Nuvens, a start up working with Lua. At the time of our interview he was again employed as a Lua programmer for a PUC project, building software on top of Kepler. He talked about Kepler with enthusiasm, mentioning among other things “the idea of domestic technology”:

Ricardo: I’ve always been following this Kepler thing and finding it interesting. Rodrigo would always pass by, like this, at PUC, and we would chat and he would tell me what was happening.

Yuri: But interesting in what sense?

Ricardo: [Long pause]. You see how things change? The idea of it being domestic technology [*tecnologia nacional*]... [pause], well structured [pause]. A proposal for an actual framework for Internet development. To compete—perhaps Rodrigo would say that it’s not to compete, but let’s say that for now—

307 See appendix C, “Original Interview Quotes,” (Silvio, May 2007, “Por puro preconceito”).

with technologies that exist out there... I found it interesting.³⁰⁸

The phrase “tecnologia nacional” is remarkably ambiguous. While it can be literally translated as “national technology,” such a translation would connote a lot more patriotic pride than the term typically implies. Products that are described with the adjective “nacional” are often understood to be local substitutes for foreign products that are either not available or more expensive. I thus use the more neutral term “domestic” to translate it.

After mentioning the idea of “domestic technology,” Ricardo quickly moves to the technical virtues of the project. I have to ask him to come back to this idea a few minutes later:

Yuri: What do you mean by the idea of domestic technology [*tecnologia nacional*]?

Ricardo: Huh?

Yuri: You said that a part of what made it interesting was this idea of domestic technology.

Ricardo: Oh, right. [Pause.] Why I find this interesting? [Pause.] It’s hard to say. It’s... [Pause.] I don’t know really. Maybe a little bit of actual patriotism too. To believe that we [*a gente*] can develop really good, world-class technology. To be used by people from all corners [of the world], and which works.³⁰⁹

Ricardo’s “we” appears to refer to the Brazilians. When I ask him who he means by “we” he confirms my guess with hesitation:

Yuri: “We” means who?

Ricardo: I can say “we the Brazilians”, or I can say, “we... PUC”, “Nas Nuvens,” “the open source community.” I don’t know, I don’t know really. It’s like this.

Yuri: But who did you have in mind?

Ricardo: I think that... [Pause.] This idea of domestic technology [*tecnologia nacional*], or let’s say developed—if only initially—by people here, in Brazil. I think this excited me. Even if—“Ok, there are people from all over the world participating.” Even better! Do you understand? There are people from all over the world offering recognition [*prestigiando*] to something that started here. [Long pause.]³¹⁰

308 See appendix C, “Original Interview Quotes,” (Ricardo, July 2007, “As coisas mudam”).

309 See appendix C, “Original Interview Quotes,” (Ricardo, July 2007, “Prestigiando uma coisa que começou aqui”).

310 See appendix C, “Original Interview Quotes,” (Ricardo, July 2007, “Prestigiando uma coisa que começou aqui”).

Ricardo struggles to bring together two seemingly contradictory ideas: Kepler and Lua as examples of “domestic” or even “national” technology, something Brazilians can be proud of, and the “global” nature of those projects. He arrives at a formulation that is similar to Silvio’s prescription for Lua’s success: technology produced in Brazil can be a cause of pride when validated by acceptance around the world.

For Everyone’s Benefit

On my second day in Rio in 2007, I went to meet Rodrigo Miranda, to catch up face-to-face on what had been happening in the year that I had spent in California. One piece of news that Rodrigo related to me concerned the recent interest that PUC had developed in Lua. In fact, Rodrigo said, there was now a project, to which I will here refer as “Iris,” aimed to promote the use of Lua by setting up a number of publicly visible Lua projects (patterned after Rodrigo’s own Kepler, described in the next chapter) and looking for funding from foreign companies and local agencies. Rodrigo talked about Iris with excitement, even as he mentioned some of the internal politics around the project and the fact that the talk had been running somewhat ahead of the action.

When we returned to the topic a few days later, Rodrigo told me that a little over a year ago he presented the idea to “Chico,” a friend of his who at the time was starting to exercise a certain amount of influence at PUC—at least more so than Rodrigo himself. Chico liked the idea and went to talk to his boss “Carlos.” Carlos had heard of Lua by then, but did not yet have a plan of how to use it to PUC’s advantage. He liked Rodrigo’s idea, however, and took it to *his* boss. According to Rodrigo, Iris eventually made it to the highest levels at PUC, and became “a big thing.” As the project grew in status, PUC managed to line up some money from “Softnet,” a large American IT company. Now that there was money involved, naturally even more people were excited.

Rodrigo said that he had discussed the plan with Roberto early in the process, but Roberto perhaps had not taken it seriously, thinking it was another of Rodrigo’s “crazy” plans, unlikely to materialize. After the project grew big, says Rodrigo, Roberto heard of it and complained about not being in the loop. While this had caused some tensions around Iris, Rodrigo was confident that this was a matter of miscommunication, and was hopeful that the issue would get resolved, as Roberto would come to see Iris for the opportunity that it was.

A few days later, Rodrigo introduced me to Chico, who greeted me with much excitement. After a brief tour of his lab, Chico talked about his efforts to “evangelize” Lua. *PUC had not been paying attention to Lua*, he said. *But it’s changing now*. At some point, he turned to Rodrigo and remarked: “I am going to go to the Dean and say: Look, there is a guy from *the United States* doing research on Lua! Why are people *here* not paying as much attention?”

Later that month, with some help from Chico, I met Carlos, who told me of his reasons supporting Lua. As he explained, he had known Roberto for some time, as a friend of a friend, and had heard of Lua before, but had not thought seriously about its potential until recently, when he

started noticing its popularity abroad. “When I took this planning role, and started looking for the potential of the university, what it had,” says Carlos, “I noticed that there was a fairly big movement actually using Lua.”³¹¹ Chico’s return to PUC brought Carlos in contact with Rodrigo Miranda and his ideas about strengthening Lua’s local position. Carlos embraced those ideas and took them to the administration of the university, making it his goal to reinforce the links between Lua and PUC, a tie that he thinks would be beneficial for both the language and the university.

Carlos: The University wasn’t really seeing the importance that Lua was having in the international context. And the members of the administration agreed with this. And I mentioned that we need to try and see what ways there could be of strengthening this association—between Lua and PUC.³¹²

Carlos refers to his efforts as “evangelization.” “You have to win people’s hearts, people’s minds,” he explains.

Stronger Lua would benefit not only PUC, but also the city and the country, stressed Carlos. It could help curb the prejudice often applied to local technology—the kind of prejudice that Silvio discusses in an earlier quote. Unlike the younger engineers and scientists Carlos was not afraid to come across as someone whose technical judgment was clouded by his concerns about the future of Brazil, speaking enthusiastically about the opportunity of local economic development that Lua created:

Carlos: The perception that if we can unlock the value [*valorizar*] of this language, it will serve for everyone’s benefit, for Rio de Janeiro and for Brazil. Because a peripheral country always has this... Always: “Oh, no, this is just made by a Brazilian.” You buy a product in Brazil, if it breaks you say: “Ah, it’s Brazilian.” You buy foreign equipment, a car, you say: “Ah, look how wonderful.” If it breaks, the person almost doesn’t even say anything. But if you bought a Brazilian product [and it breaks], you say: “That’s because it’s Brazilian.” And this applies to software too. Despite our great position in banking software, in the financial sector, etc... But Brazil is not very aggressive in this *off-shore* field.³¹³

According to Carlos, Lua could change this perception, leading to new economic opportunities for the region.

This change, however, would require support from PUC’s administration, policy makers and funding agencies, explained Carlos. Local companies were unaware of the opportunity that Lua offers them, quite likely due to the same prejudice. The few that understood this value found themselves unable to find Lua programmers. (Carlos seemed to suggest that the fact that local companies did not use Lua then in turn means that there are little incentives for the programmers

311 See appendix C, “Original Interview Quotes,” (Carlos, April 2007, “Percebi que existia um movimento”).

312 See appendix C, “Original Interview Quotes,” (Carlos, April 2007, “Evangelização”).

313 See appendix C, “Original Interview Quotes,” (Carlos, April 2007, “Em benefício de todos”).

to learn it.) This could be changed with some support. “And what I also presented is that maybe we need to make an effort to look for resources,” said Carlos. “To bring financing agencies here, tell them: ‘Let’s train Lua programmers.’”³¹⁴ Such training programs, combined with other forms of support to companies interested in offering Lua services to foreign clients could jump-start a new sector in Rio IT market.

The initiative was to be presented a month later in a meeting with the Secretary for Economic Development of the State of Rio de Janeiro:

Carlos: ...where one of the proposals that we brought for him for the development agenda would be the issue of information technology, but with a focus on Lua, as Brazil’s differential for *off-shore*. Java could be in China, in India... India has this ease with language, which here in Brazil—perhaps we do not have that. But without any question, Lua is a differential where today we have the conditions to quickly form a critical mass, if we can articulate this. [...] We’ll invite financing organs, in the area of science and technology, the National Bank of Economic and Social Development, so that they come here, and bring the companies that are already in the process of religious conversion [*catequese*]... [Laughs.] For them to also talk about their interests. In other words, you would in a way be underlining and certifying that this is a Brazilian product, open source, and even [showing] this creativity here in Rio de Janeiro.³¹⁵

With some support, Rio software companies could start offering outsourcing services in Lua, benefiting from their proximity to Lua’s base. As Lua would grow further (in part thanks to this local business activity) and as long as its national origin is properly highlighted, Lua’s success could help change the perception of Brazil and Rio by displaying the technical creativity that can be found in Rio de Janeiro.

While wanting to position Lua as “a Brazilian product,” Carlos was also global in his thinking, feeling that Lua and PUC needed global partners to make Lua a true success.

Carlos: And also to start looking for large partners, right? Because if Lua has a great reach at the international level, if it has a series of qualities that the community recognizes, why not attract heavy-weight partners, big international *players* [English], like Softnet, IBM, to talk about this topic.³¹⁶

As a step along this plan, Carlos had recently negotiated, on remarkably good terms, research and development funding from Softnet, which was to go towards projects using a combination of Lua and Softnet’s technology.

When I discussed the Iris project with Roberto Ierusalimschy, I learned that he was less

314 See appendix C, “Original Interview Quotes,” (Carlos, April 2007, “Programadores de Lua”).

315 See appendix C, “Original Interview Quotes,” (Carlos, April 2007, “Uma proposta”).

316 See appendix C, “Original Interview Quotes,” (Carlos, April 2007, “Parceiros de peso”).

then excited about the idea. While this had partly to do with his fear that Iris would end up competing with his own research group for “Lua” funding,³¹⁷ he also feared the potential costs to Lua itself. “I’m not sure whether PUC has a clear notion of what it wants to do with Lua,” he explained. Even with the best intentions, the project could harm Lua.

Roberto: For instance, because legally, they... PUC is the copyright owner of Lua. So for instance, if they write any contract binding Lua—the name “Lua”—with something... For instance, I don’t know, but for instance IBM³¹⁸ presents: “Oh, I’m very excited with Lua! Let’s do an exclusive contract for something, something about Lua.” And then: “Oh, it happens that there is a contract now between PUC who is the legal owner of Lua and IBM that such and such development around Lua only can be done inside this.”

PUC had absolutely no experience with free / open source software, explained Roberto, and hardly much experience with software licensing in general. In other words, lacking understanding of how free software worked and how Lua fit in the world of free software (an issue that Roberto himself felt he barely comprehended, as we will soon see) PUC’s administration risked creating the wrong kind of ties between Lua and the local context which it was finally starting to separate itself.

Among other things, such contracts would highlight the image of Lua as software from “a South American country”—a product potentially enmeshed in the bureaucratic complications that one could only successfully navigate by having strong local connections.

Roberto: For instance, that could be very bad for the image of Lua. Suddenly a lot of people in the world are using Lua and there is a risk that IBM could accuse them: “Oh, you cannot do that because it’s under a contract between me and PUC...” This is a very, very big risk...

Because one of the main problems Lua had... I think now finally it doesn’t have it anymore, was that people were very unsure about using a software from... as you say, from a South American country, they don’t know how we think and things like that. So they are afraid, “Can we use Lua? Is it really free? What they are going to do?” Things like that. *You* told me, you know that.³¹⁹ And then after all these years we kind of conquered some kind of credibility. “Oh, ok, Lua seems to be something stable. A lot of people use it, so I don’t think... there is not going to be any problem.” So more and more people are starting to use Lua. And more important, more and more people are starting to *admit* that they use

317 Roberto: For instance, Iris submits a project to somebody. And for the agency, this is “Lua at PUC.” So [for them it’s]: “I gave money for Lua at PUC.” And then there is another project, our project for Lua at PUC: “Oh, we already gave money for that, we are not going to give money again for the same stuff.” Outside, they are not going to see the difference.

318 Roberto appears to use IBM as just a hypothetical large company. In particular, IBM is not “Softnet.”

319 Roberto is referring to our discussion of my interview with Rich, quoted earlier in this chapter.

Lua. And then suddenly it transpires that there is a hidden contract done in South America between IBM or Softnet or Microsoft or whatever and PUC, which is the legal owner of Lua, that... whatever it is in the contract [laughs], it doesn't matter very much...

Since Lua had been released under an open source license (with the tacit agreement of the PUC administration), the question of who owns Lua's copyright could turn out to be quite irrelevant.³²⁰ The *uncertainty* created by badly written contracts (or simply bad publicity around contracts written well) could be enough to scare off Lua's users. "Nobody wants to go to court in Brazil to try to use Lua," said Roberto. "Even if they know 'Oh, I'm almost sure I'm going to win that.'"

Concerned with unlocking Lua's value and using it for everyone's benefit, Carlos wanted to "reinforce the connection" between Lua and the local context in which it was born. This association, however, was precisely what had to be undone for Lua's global success. As Silvio said, to succeed in Brazil, Lua needs to first succeed abroad. To succeed abroad, however, it had to avoid any associations that could go with "software from South America." Such successful detachment, however, reduced Lua's ability to change the perception of Rio and Brazil. "What I would want to happen would be that PUC got a better notion of the outside value of Lua before a notion of the inside value," Roberto said. PUC could do little to enhance Lua's global position, but could do much to hurt it.

When I returned to Brazil in the end of 2008, the conflict around Iris seemed to have been largely worked out. Carlos appeared to have understood some of the challenges faced by Roberto—the challenges to which I turn in the next section. He started consulting Roberto more closely on PUC's approach towards Lua and the ambitions of the Iris project were largely scaled down.

Reading Smoke Signals

Despite Lua's permissive license and the active interaction between the authors and the Lua community, Lua never fully moved to the open source development mode: all changes to the Lua code itself have always been made by the three members of the Lua team, mostly by Roberto Ierusalimsky. Roberto explains this decision in part by the need to keep the language small:

Roberto: Yeah because I think most other programming languages—open source—they are much more open than Lua. So they are... For instance, in Python or Perl, you have a lot of people that actually vote for changes and there are those kind of open decision, open-source decision-making strategies and things like that. You can enter as a committer and you are promoted as a developer and then you have the right to go and there is all this hierarchy. And Lua is just the three of us... [Laughs.]

³²⁰ As with other open source projects, the only real property involved is the trademark for the name "Lua." Others can release a modified version of Lua, but they would have to call it something else. (The Lua trademark belongs to PUC.)

Yuri: Have you ever thought of doing that?

Roberto: No. I mean, in what sense? We thought about *not* doing that. [Laughs.]

Yuri: Ok, but you're saying this is a possibility that was discussed and you decided not to?

Roberto: [...] Oh yeah, yeah. We decided not to... [Laughs.] [...] Because exactly... for several different reasons. One is exactly that I don't think that this is necessarily good for a language. I think that always it tends to grow. It's very difficult for people to agree on what to remove from the language so I think this is one of the main points.

As Roberto explains (and many users agree), open source development works well for *adding* features but makes it harder to control the growth of the language, not to mention *removing* features. Since Lua is generally recognized for its minimalism and small size, it may in fact benefit from the “cathedral” rather than the “bazaar” development model (see Raymond 1999).

Roberto then adds another reason, however: his desire to maintain control over Lua's future.

Roberto: The other main point is that we really like—I mean, someone once said that in kind of a very aggressive way, not that aggressive, but... It's our language, I mean, we like doing it and it's...

Yuri: Someone said what? That it was your language?

Roberto: Yes, but it was kind of “Nobody has nothing to do with that, it is his language, he does whatever he wants and he doesn't care what people think about it” [says imitating aggressive voice] and I mean, I care a lot about what people think about it, but I really want to keep this privilege of—this is the language I developed, I want to have the language the way I want it.

While Roberto's desire to develop Lua as he wants it could be understood as a case of an artists' concern for the purity of the work, it connects to a concern that comes up frequently in the interviews: the risk of losing control over Lua.

There are two sources of this concern. One is the inherent uneasiness of Lua's position as an international programming language with a base in the wrong place— “software from South America” trying to make it globally. The other is the Lua team's limited knowledge or understanding of Lua's use abroad, and especially of the social dynamics of that use.

At the beginning of our first 2007 interview, Roberto suggested starting with a different question from the one that I first asked him:

Roberto: I think maybe start a little earlier? Because this is something that I was thinking today and something that I am always thinking. The main point is that

we have a very, very rough idea of the growth of Lua and how Lua is being used and things like that. We are always kind of... I don't know if this is because we are in Brazil or if it would be the same if you were living in Silicon Valley, but my impression is that I always kind of try to read smoke signals to try to realize that there is a real growth, there is no real growth.

While Roberto says this issue is always on his mind, it turns out that a specific event has led him to spend time thinking about the topic earlier that day.

As Roberto proceeded to tell me, over the course of recent months (from November 2006 to March 2007), Lua had enjoyed a dramatic change in its position in TIOBE TPCI—a popular ranking of programming languages based on Google queries.³²¹ After spending a long time in the group of “next fifty” (languages that TIOBE rates as among “the top one hundred” but not to “the top fifty,” without assigning them individual rankings), Lua entered the top fifty in December 2006, and then started a slow ascent within this group. The March 2007 ranking came out the day before our interview. Lua had made a dramatic jump: to the twenty fifth position. Since most of the software development is done in just a small number of programming languages (see chapter 2.2), being in twenty fifth position did not imply huge market share and did not yet qualify Lua for TIOBE’s “A level” designation. However, it most certainly put Lua on the map, leaving it steps away from the doors of the most exclusive group in software: “major” programming languages.

This new success was so substantial, that Roberto laughed in disbelief as he talked about it:

Roberto: [Laughs.] I am not sure if the index is wrong, I mean, it's a very, very big jump. It jumped over twenty other languages in one month. Very strange. [...] It's very strange, I'm not sure if they are right. But the main point is I have no idea how we are climbing up, what happened in the world that put us that much, I mean... We have a thin idea that there is a new book about Lua that is *Beginning Lua Programming*. Actually, this is the first, how can I say, the first outside book that is really, really about Lua.

The reasons for Lua's recent success were largely a mystery to Roberto, who had to read “smoke signals” from a distance to learn whether Lua was growing and what factors might have been contributing to its growth. He similarly had little idea if Lua's use will continue to grow, how quickly, and if so, what this growth would bring.

Apart from the discussion on the list, publications and blogs are the main source of information. “You're always trying to understand what is going on,” he says laughing. Roberto relates a story about Cameron Laird, a well-known columnist writing about programming languages development, mentioning Lua in his column in 1998.

Roberto: And then in '98 in one of his columns he wrote about Lua: “It's a very

321 <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

nice language, still has a very, very small base of some tens of thousands of users.”³²² At that time for us—in ’98—to learn that we had tens of thousands of users! We were discussing it: if we have one thousand users, maybe... “Let me see, I think we have a few hundred users, maybe one thousand.” And then this guy says: “They have very few users, just tens of thousands.” We were: “How?..” We are always kind of... [...] I try to *acompanhar*, to follow blogs, and I have a link in my bookmarks. There is a search in Google blogs for “Lua” and “programming” or “games” and every day I go to see news about Lua in the blogs to try to have a feeling if there is something new happening or things like that. And sometimes I see: “Oh, this is interesting.” But sometimes I see something that I get surprised: “Oh, there is...” It’s difficult to...

Following Lua news from an office in Rio de Janeiro takes some work.

While Roberto wants Lua’s popularity to grow further, this difficulty of understanding the factors underlying this growth makes this growth itself a source of uncertainty. Lua community members often mention the many occasions when Roberto had said that he would perhaps prefer Lua’s popularity to grow *slower*. For example, after Lua’s entrance into “top fifty” in October 2006, one of the list members asked: “How long before Lua gets into the top 20, I wonder?” Roberto replied to the list: “Veeeeeeeeeeery long, I hope ;)”³²³

While such uncertainty is inherent in development of free software (since its use cannot be easily tracked through sales), especially when it is done on a small budget without the possibility of expensive market research, Roberto’s situation is made more serious by his relative isolation from the place where his programming language is used most actively. Apart from the interactions on the list and the occasional Lua workshops (which happened in 2002, 2005, 2006 and 2008, three times in U.S. and once in the Netherlands³²⁴), Roberto has little contact with the programmers that use the language that he designed.

This lack of face-to-face contact (and the mystery that results from it) is illustrated most starkly when Roberto talks about one of the list members, whom he mentions as a source on influence on Lua’s design:

Roberto: Yes, in the list there are some people that... At least Joe Simon [pseudonym]. Joe is someone I really respect. This guy—it’s unbelievable again.

322 “Lua is slimware; all the source code totals well under 10,000 lines. Its user base is also small; there might be only a few tens of thousands of Lua programmers in the world. They’re very fond of this language, though, and the imminent explosion of ubiquitous embedded processing (computers in your car, in your plumbing, and in your kitchen appliances) can only work in favor of Lua.” (Laird & Soraiz 1998.)

323 In the months that followed this interview, Lua inched its way up the rankings, entering “top 20” in July, taking the 18th position. It has since maintained a position at the bottom of top 20, moving only slightly up or down every month.

324 The first of those workshops (at Harvard) focused narrowly on improvements to Lua’s API and is typically not counted as one of the “Lua Workshops.” The other three (in San Jose, CA in 2005, in Venlo, Netherlands in 2006, and at George Washington University in Washington, D.C. in 2008) were aimed for a more general discussion of things happening in the Lua community.

Luiz Henrique jokes³²⁵ that this is really a group of ten people, because it's unbelievable! He knows everything, he answers everything in fifteen minutes... The most exotic questions about operating systems, about... A few days ago, someone asked something about Microsoft email program [Outlook] [...], and I never heard about... he answered in ten minutes: "Oh, you go to that menu, you use that option, and then that solves the problem." How does he know that?! [Laughs.] And he knows everything about operating systems, about architecture: "The Pentium 4 has a specific hardware circuitry for that operation, but in the other one it gets off the other, and then the cache..." He knows... It's unbelievable how much he knows.

In addition to his incredible ability to answer the most obscure questions, Joe has developed a tool used by many Lua users. Despite those contributions, Joe remains a mystery to Roberto, who has never met him.

Roberto: That's one of the reasons we think he's ten guys. [Laughs.] We never met him. He's very secretive. I never saw a picture of him, he doesn't have a home page.

Joe, whom some Lua users assume to live in Europe (based on nothing but his email address), is "secretive" according to Roberto and has not attended any of the Lua workshops despite his active involvement with the language. He may well have remained a mystery to Roberto even if Roberto's home was in Palo Alto, California. Being in Brazil, however, ensures that most members of the community have much of the same mystery that surrounds Joe's persona.

To understand the community around Lua, Roberto finds that he has to get a better understanding of the "culture" of open source. He does so in part by reading a book about open source:

Roberto: Yeah, because there is this *culture*. For instance, I have this... I just... [Rodrigo] Miranda gave me a copy of this book called [...] *How to Run an Open Source Project*.³²⁶ It's assumed that an open source project is something that is open source decisions and... It even considers the possibility of... what they call... [pause] "the benevolent dictator." Is that the name? But it must be *benevolent*, that dictator. [...] And it's exactly because of that, he said in the book, because there is always this... the possibility of fork if the dictator is not benevolent, or if it's *really* a dictator. People will fork to another project and so... I mean, you cannot be a dictator, so I mean. So he puts that as almost a result of an open source project—that it will have an open source development, decision-making organization.

The book assumes, says Roberto, that the development of an open source project is driven by

325 Luiz Henrique points out that the joke was actually made by someone else.

326 Fogel (2005), *Producing Open Source Software: How to Run a Successful Free Software Project*.

what Roberto calls “open source decisions”—a communal decision making process in which no single person can make unilateral decisions. Projects that do not adhere to those norms run the risk of “forking.”

The book explains forking and its effect on the decision making the following way:

The indispensable ingredient that binds developers together on a free software project, and makes them willing to compromise when necessary, is the code’s *forkability*: the ability of anyone to take a copy of the source code and use it to start a competing project, known as a *fork*. The paradoxical thing is that the possibility of forks is usually a much greater force in free software projects than actual forks, which are very rare. Because a fork is bad for everyone [...] the more serious the threat of a fork becomes, the more willing people are to compromise to avoid it.

Forks, or rather the potential for forks, are the reason there are no true dictators in free software projects. This may seem like a surprising claim, considering how common it is to hear someone called the “dictator” or “tyrant” in a given open source project. But this kind of tyranny is special, quite different from the conventional understanding of the word. Imagine a king whose subjects could copy his entire kingdom at any time and move to the copy to rule as they see fit. Would not such a king govern very differently from one whose subjects were bound to stay under his rule no matter what he did? (Fogel 2005, p. 88.)

When software developers release code under a free software license, they do not simply share the *product* of their work. By sharing the source code and by growing a community around it, they share *nearly all there is to the project*. A free software license allows any recipient of the code to not only use or share it, but also to *modify* it as they see fit. Having modified it, they can make a case to other users that their version is better than the original. If the leaders of the project reject the modifications, the users can still make a choice to use the modified version.³²⁷

Convincing the user community to abandon its current leader is a hard task for the claimant, as it requires, at the very least, a combination of demonstrable technical prowess and communication skills, lacking which the claimant will likely be ridiculed.³²⁸ (One derivative of Lua was subject of such ridiculing by the mailing list members on a number of occasions in 2007 and 2008.) Those who create forks thus typically present them as subordinate projects, downplaying their significance, for instance by calling them “experiments.” Over the years, the members of the Lua community have undertaken a number of such “friendly” forks. Most of them, says Roberto, had a positive effect on Lua, as they either helped develop new ideas later

327 In fact, as Raymond (1999) points out, in a section called “Promiscuous Theory, Puritan Practice,” the right to create forks is essential to an any open source license and such licenses can be said to “implicitly encourage” forking. At the same time, he argues, forking rarely happen in practice because of “an elaborate but largely unadmitted set of ownership customs” and a set of “taboos.”

328 Such skills represent the minimum requirements, since a key extra factor is usually needed: a justification for what is often perceived as a clear violation of a social norm.

incorporated into Lua or demonstrated certain proposals to not be a good idea. The possibility of a successful adversarial fork that leaves the project leaders without the community, however, is ever-present. It looms especially ominously for leaders who may trust their own technical instincts but not their grasp of the *culture* of their users—or find themselves unwilling to accept what they see as the demands of this culture.

Roberto: Yeah, I learned a lot about exactly... For instance, I didn't know that there... is so strong this culture, of exactly what the meaning of "open source" is for some people. I thought that Lua of course was open source. Now I say: "Lua is open source but it may not be what you think it means to be open source."

After reading the book, Roberto decided to take the risk and proceed with what he feels is best for Lua, disregarding what he understood to be the author's advice. He remained worried about the consequences of that decision.

Roberto's experience with open source illustrates the challenges that open source presents for software developers at the periphery of the software world. While open source software development presents many opportunities for such participants, producing open source software successfully requires *higher* levels of competency in the software culture than other forms of engagement in software development. The developers working for Alta (chapter 3.1), who produce commercial software for their local clients, must only project a competent image to their clients and local peers. Their clients typically have limited grasp of the software culture and, in most cases have few options for looking for expertise outside Brazil or even Rio de Janeiro. The authors of Lua cast their lot with a community of developers based largely outside Brazil and comprised of many people who are *more* fluent in the software culture than Lua's authors themselves. A firm grasp on the software culture becomes crucial for success.

Ginga

When I returned to Brazil at the end of 2008, I learned of a new development that promised to substantially increase Lua's use in Brazil: the language had been included in the Brazilian digital television standard. In the short term, this meant that companies interested in selling receivers for digital television signal in Brazil ("set-top boxes") had to support Lua in their products. In the longer term, this promised the creation of a local labor market for Lua programmers in developing content for interactive television. While I heard of such plans during my visits to Brazil in 2005 and 2007, many of my interviewees seemed to dismiss them as just talk. At the end of 2008, however, Lua's place on Brazilian set-top boxes seemed all but assured.

Lua's inclusion in Brazilian digital television standard did not come from specific efforts of the Lua team. Rather, Lua entered the standard as a component of "Ginga"—a digital television system developed at PUC-Rio.³²⁹ From the late 1980s, Luiz Fernando Gomes Soares, a

329 To be more precise, Ginga represents a particular layer in the digital television system, typically referred to as "middleware."

PUC professor who once served as the Master's advisor for Roberto Ierusalimsky, had worked on the problem of representing the structure of multimedia presentation. In the late 1990s, this work resulted in NCL ("Nested Context Language")—an XML-based language for expressing hypermedia presentations, similar to a popular format called SMIL, but offering a different way of representing temporal relations between multimedia objects.³³⁰ While NCL did not get adopted by the foreign industry as Lua was, Luiz Fernando and his collaborators published a large number of papers on their research, many of them in foreign computer science conferences.³³¹

In 2004, Brazil's government announced a call for research proposals related to the future Brazilian Digital Television System (SBTVD). While the lower layers of the system (at the level of signal modulation) were to be based on the system earlier developed in Japan (ISDB), the government was looking for local innovation, in particular at the higher level of interactivity. Luiz Fernando's laboratory submitted a proposal and NCL was accepted as a research project and then as a basis for a "middleware" of SBTVD under the name "Ginga," in reference to a basic move in *capoeira*.³³²

As with Lua, foreign success has been crucial in making NCL-based middleware accepted by the local industry, says Luiz Fernando.

Luiz Fernando: The first big difficulty that we had was that people didn't believe that we had actually built something much better than what was being proposed abroad. It was very interesting that Brazil would not believe it and it took people abroad to believe and say so first, for Brazil to later accept it.³³³

Foreign approval of NCL, expressed in publications citing the work of Luiz Fernando and his colleagues helped gain local support.

NCL is a "declarative" language, which focuses on describing the multimedia objects and the relationships between them. To be truly interactive, however, the system needed a procedural language that could be used to program the behavior of the objects. Java was an obvious candidate. "Java is a language that is used by practically all TV systems, digital TV systems," says "Michel," one of the Ginga collaborators at PUC. "So, to maintain compatibility, Brazil

330 The most important consequence of the difference in temporal semantics between NCL and SMIL is that NCL makes possible live editing of the presentation. That is, the editing of a SMIL presentation must be completed before the presentation is viewed, while NCL makes it possible to edit the later parts of the presentation while the earlier parts are already being displayed. This is ability is important for live digital television.

331 For example, Casanova et al. (1991), Santos et al. (1998), Costa et al. (2006).

332 Ginga is designed to work with terrestrial television, that is signal that is sent by an antenna on the ground and picked up by a simple television antenna. Different countries use different several standards for digital terrestrial television (DTTV). Brazil is basically introducing the Japanese system, though adjusted for higher resolution. However, national DTTV systems in current use do not provide much interactivity. The interactive televisions systems that exist (e.g., those provided by private cable companies in the US) typically require a "return path," that is some form of broadband Internet connection (cable, DSL) or telephone. Ginga offers a degree of interactivity without a return path: the code is transmitted by air and is executed on the set-top box. Similar solutions have been proposed, but Ginga is actually the first open implementation.

333 See appendix C, "Original Interview Quotes," (Luiz Fernando, January 2009, "O Brasil não acreditava").

decided to also have a Java API in its middleware.”³³⁴ A research group at the University of Paraiba, headed by a PUC-Rio graduate, had already done much research on putting Java into set-top boxes. The two universities started working together, developing a system that combined NCL and Java.

PUC researchers, however, decided to also add Lua. Michel points out two reasons: the personal connections to the Lua group (located in the adjacent building) and Lua’s obvious fit for the task.

Michel: Well, Lua comes in first of all because it was easy for us to work with the Lua community, which also is a language developed here at this university. Second, because we started seeing that we needed for the terminal, for the set-top box, something that would be light, that would be efficient, that wouldn’t occupy much memory. And we started to notice in the comparisons done by third parties, sites that do comparisons of scripting languages, that Lua is very well received for those characteristics, for being a light and efficient interpreter. [...] With digital TV, people soon start thinking of entertainment, of applications, games, those sort of things. Those are interesting applications to start thinking about, running them on the receiver. Lua being much used in the game industry, this again had much to do with it. So, it was because the sum of those characteristics.³³⁵

Indeed, set-top boxes provide a context in which Lua is strongest and quite similar to the context where Lua is used most often—computer games. Companies outside Brazil include Lua in their set-top boxes without any government mandate.³³⁶

Michel later offers a different explanation, however: there was little cost to including Lua:

Michel: Java [...] has a heavy side in terms of the virtual machine. But if you don’t put Java, you already limit applications. The broadcasters, they are worried also about importing programs, with compatibility. So, if you remove Java from the system, you already exclude a number of interactive applications. And Lua is something that doesn’t weigh anything. So, whether you put Lua or not put Lua... there is practically no advantage to not putting Lua in. So, if there is already something that can do work well, that is already integrated into NCL, there is no harm in adding it. It’s very light, just 50K. [...] So, it’s there as one more option.³³⁷

While NCL was already struggling for acceptance as a local technology, inclusion of *another*

334 See appendix C, “Original Interview Quotes,” (Michel, July 2007, “Pra manter interoperabilidade”).

335 See appendix C, “Original Interview Quotes,” (Michel, July 2007, “Usado por essa indústria de games”).

336 The examples include Verizon FiOS (<http://www.verizon.com/fios>, see Perez 2009), Active-TV (<http://active-tv.org/>) and Vudu (<http://www.vudu.com/>, see Hessel 2008).

337 See appendix C, “Original Interview Quotes,” (Michel, July 2007, “O ganho de não botar Lua”).

local technology did not burden the proposal too much. Lua was just “one more option.”

When I arrived for my interview with Luiz Fernando in January 2009, I found him in the midst of a meeting. As it turned out, he was meeting with representatives of a company that wanted to license PUC’s implementation of Ginga to offer services to foreign hardware manufacturers interested in selling set-top boxes in Brazil.³³⁸ All of those set-top boxes were to run Lua. As I waited for the meeting to end, I passed my time looking at the poster on the adjacent door—it advertised the course on NCL programming, which was offered at PUC in the Fall of 2007 and included an introduction to Lua. The existence of “a Lua course” at PUC was an important news, relayed to me diligently by a number of my former interviewees. The course used the Portuguese translation of the Lua manual, completed by one of Roberto’s Ph.D. students in August 2007. After the interview, I passed by LabLua to chat with some of Roberto Ierusalimschy’s students. I saw *Programming in Lua* open on one of the last chapters. The same student who translated the Lua manual was finishing the translation of the book.³³⁹ The translation work was not related to Ginga—it was paid from Kepler’s grants. Local efforts to promote software based on Lua, however, seemed to start adding up. Late that day I gave a well-attended talk about my own Lua project, organized by Chico, whose research lab, dedicated to running Lua on small electronic devices, had expanded to several rooms. Having gained success abroad Lua was starting to appear ready to come home.

338 PUC’s implementation of Ginga is available under GPL license. This means, that companies can use it without paying royalties to PUC, but are required to release their implementation under the same license (if they were to distribute the software). Those who do not want to do so have an option of obtaining the software from PUC under a different license but for a fee.

339 The draft of the translation was completed later that month.