Lyee Internet Information

Interview

with

Prof. Mohamed Mejri at Canada Laval University

Mr. Fumio Negoro, president of The Institute of Computer Based Software Methodology and Technology

Prof. Mejri

Mr. Negoro

Date:	Thursday, November 15, 2001
Place:	Head office of The Institute of Computer Based Software Methodology and Technology
Moderator:	Prof. Hamido Fujita at the Iwate Prefectural University

【Profile of Prof. Mohamed Mejri】

1968 Born in Kélibia, Nabeul, Tunisia

1995 Engineering Degree in Computer Science with General Honors. University of Computer Science (ENSI), Tunisia

1998 Master Degree in Computer Science with General Honors. Laval University, Canada

2000 Awarded a Ph.D. in Computer Science with General Honors. Computer Science Department, Laval University, Canada

2001 Appointed Professor in Computer Science. Computer Science Department, Laval University, Quebec, Canada

Present ditto

[Principal Research Interest]

- Programming languages, computer security, formal methods, software engineering

[Principal Publication]

- M.Debbabi, N. Durgin, M. Mejri, and J. Mitchell. Security by Typing. Accepted for publication in the International Journal on Software Tools for Technology Transfer (STTT). Springer Verlag, 2001
- K. Adi, M. Debbabi and M.Mejri. A Logic for Specifying Security Properties of Electronic Commerce Protocols. Accepted for publication in the International Journal of Theoretical Computer Science, TCS'2001.

Prof. Fujita We are delighted to learn that Prof. Mejri of Laval University is going to join the International Scientific Joint Research Project. Welcome you to the project, Prof. Mejri. Our interview usually begins by asking an interviewee about his/her background and research interest. Would you

briefly introduce yourself?

Prof. Mejri **I was awarded an engineering degree in computer science in Tunisia in 1995. It took six years. From 1996 to 1998, I was on a scholarship from Canada to get M.A. Immediately after I was awarded M.A., I entered the Ph.D. program of the same university, Laval and was awarded Ph.D in 2000. In 2001, I was appointed professor at Laval University, and am still working there with the same position.**

My research subject for M.A. and Ph.D. was security of computers. In particular, I worked on analysis and verification of e-commerce protocols. This research subject is deeply and closely associated with various fields of computers such as programming languages and formal methods. To tackle a security problem, we also have to get involved in programming languages and formal methods.

Currently, I am a member of the LSFM group, which stands for Language, Semantics, and Formal Methods. As the name indicates, I am interested in languages and formal methods. Provided, however, my interest in these areas is stirred up only in the context of security issue.

Right now, I have several M.A. students who are working with me. They are also researching into the security.

I also want to mention that we conducted several other collaboration projects on security with other universities and groups. For example, I worked with John Mitchell of Stanford University and with the National Defense Research Group in Quebec. Through this project, we conducted static and dynamic analyses on the programs and produced very interesting results.

Prof. Fujita How did you become acquainted with Lyee, then? I would also like to know what brought you to Japan this time?

Prof. Mejri **Both Prof. Bergeron and Prof. Mourad of Laval**

University belong to the LSFM. First Prof. Mourad talked about Lyee to me. I was struck by a feeling that there is something interesting, but at the same time, I have many questions including Lyee's philosophical perspective. Later on, Prof. Bergeron gave me an additional talk and documents on Lyee.

Given the information I gained from these two professors, some image on this methodology was gradually formed in mind, although I had some questions. These two fellows focused on a technical side of Lyee, particularly on legacy conversion, e.g. from assembler to COBOL. Even though I did not know well Lyee's metaphysical part, I understood a high level of automatic conversion would be possible with Lyee.

During the process of familiarization of Lyee, my feeling shifted from one point where Lyee is somehow interesting to another where Lyee is indeed an in-depth methodology sustained by a philosophical basis.

After my participation in the International Scientific Joint Research Project is formalized and continues to study Lyee, I suppose that I will discover more interesting ideas which I am not aware of now. It is true that I was first interested in a technical side of Lyee, but I would also like to examine the theoretical part Lyee.

Prof. Fujita Prof. Mejri is a last visitor among would-be participants of the project. But thanks to Prof. Bergeron and Prof. Mourad, you were already well informed of Lyee and had a positive image on Lyee before coming to Japan.

We are therefore interested in what Prof. you will do in the project as an individual researcher and in how you will be involved in the project as a research unit at Laval University.

Let me go on to the next question. Would you tell me how you see Lyee now? Please give us a specific vision of the research subject on Lyee, if you have any.

Prof. Mejri **One of the most important points of Lyee lies in the fact**

that one can use the Lyee method to produce marvelous results, even though he/she does not know its philosophy to back up the entire methodology.

Let me then enumerate distinguished features of Lyee. One of them is that programs are automatically generated. This is very important because it can reduce a burden of programmers. In a way, the appearance of computers created a job of programmers. In turn, however, automatic program generation will enable computers do the programmer's job. This is remarkable. Human errors will be diminished as a result. Programming work will be simplified with the Lyee method so that knowledge and academic high degrees of computer science will no longer be needed. I confirmed this when I joined an inspection tour through Tama software factory this morning.

Second point is concerned with validity checking and verification, namely testing. We have tried numerous methods to solve this problem. To be frank with you, I do not know yet how little the Lyee method does require testing now. Absolutely no testing? Or ten percent less than that with ordinary methods? I cannot say anything certain unless I study more about this issue, but still the claim of no testing with Lyee is still a very important point. Third point is about maintenance. It is easy to understand that maintenance will become an easy task with the Lyee. The reason for this lies in that given definitions and conditions of requirements, programs can be generated. There is no logical aspects involved in programming. Let me add another view on maintenance. Usually, maintenance is one of the most troublesome work in the system development. Although it depends on the type of applications, more than seventy percent of the entire work needed for the system development accounts for maintenance. If this part is to be simplified, it naturally follows that time

and cost will be markedly shortened. I think this is indeed Lyee's prominent contribution to the software field.

These are the features that I know of Lyee at this stage. But its philosophical aspect has yet to be clear to me. I know it is quite essential to understand Lyee and the understanding of this part will give more suggestions and answers to other issues and problems. In other words, if one understands Lyee's theory or philosophy, he/she can easily find out other features. More specifically speaking, one will surely give a clear reasoning for no testing.

Since Mr. Negoro spent enormous amount of time on establishing the methodology with fine details so that it may be taken for granted that I cannot fully understand it for a short period of time. But I wish we could discuss the theory of Lyee sometime so that ambiguous and uncertain parts of Lyee will be clearer. One of the examples of uncertainty is what is the most suitable are for Lyee. I was told that any sort of applications can be applied with Lyee, but I have not got that feeling yet. One thing for sure is that Lyee is quite suitable for business applications. But I want to investigate and confirm for myself that Lyee may be also fit to mathematical applications.

Prof. Fujita I think this topic is worth discussing. Let us elaborate on it together. Any methodology could be discussed from a viewpoint of its most appropriate area. I believe that this issue should be discussed in its relationship with applications.

While Mr. Negoro was developing Lyee based on the theory and philosophy, he was engaged with development of business applications. I suppose that his experience through business development helped him clarify and establish the methodology. Against this backdrop, Lyee may look more suitable for business applications than for

others.

However, the next issue will be what kind of adjustment should be made when Lyee is applied to other applications but businesses.

As for business applications, you have already a certain standard for them. In a sense, you become an expert of this area. I guess by adapting this standard to other areas in a various manner, you could develop other types of applications.

However, you have not built expertise or know-how on such areas as mathematics, which Lyee has not experienced. Real-time system is one of them as well. Even in this area, once a mathematical model is built and applied to other areas, automatic program generation will be possible. I believe this is such an interesting topic that we should consider it as one of the main subjects to the International Scientific Joint Research Project.

Mr. Negoro **I have found that Prof. Mejri's comments are interesting and striking the core of Lyee. He listed four characteristics of Lyee, but I would like to add one more point. That is, development work can be done in parallel.**

Besides these features, I should not forget to mention that Lyee's methodology incorporates semantics into it. With Lyee, semantic testing is replaced by a certain way of syntactic checking. I have not given a sufficient explanation on this relationship, and I know that some parts remain unclear. However, conventional methods do not consider semantics, and did not provide any proper definition of semantics.

As far as suitable areas for Lyee are concerned, an aptitude issue is always brought up for any methodology, as Prof. Fujita mentioned. But what I want to emphasize here is that a problem of use of computers should not be mixed up with a software problem. Lyee's software structure is not completely

friendly with current computer architecture. In this sense, there is no computer architecture which is completely fit to Lyee's structure. I sincerely hope that this will be soon resolved as an issue of engineering.

One more important point is that Lyee's structure does not allow viruses to reproduce themselves.

Last but not the least, I want to say I was very much pleased to hear Prof. Mejri's commenting that if you understand the idea behind Lyee, you will understand Lyee's other possibilities. I believe that is absolutely right.

Prof. Mejri **I have just forgotten to mention the work in parallel. I also recognized it as one of Lyee's features that Mr. Negoro pointed out.**

I also agreed with you, Mr. Negoro, in that computer's architecture should be modified. A current architecture was built quite a long time ago so that it is not suitable for our needs of the 21st century. It is in a sense surprising that nobody did try to create a new architecture which meets our current needs.

You said that Lyee could be applied to any sort of applications. I suppose it is true, but I simply cannot see it clearly at this stage. The reason for this is that a certain way of thinking has already been embedded deep in my mind. That requires me to develop an idea in a logical manner. I guess it is next to impossible for any human to take away a certain pattern and acquire a totally new way of thinking within one day or even a week. A step-by-step learning is necessary for me to understand it.

When I visited Tama Software Factory this morning, I found that a little knowledge of computer science, that is knowledge of what is programming or what is algorithms, does not really help to understand Lyee.

I want to add a few comments on the application issue. Prolog is, for example, quite suitable for artificial

intelligence areas, but it is difficult to use it for business applications, although it can be surely applied. As this example suggests, however, each methodology or language has its stronger and more appropriate area. I am keen to figure out which area is Lyee's best suitable one.

Mr. Negoro **I think it is taken for granted that compilers have their suitable fields, but methodologies should not be discussed from a viewpoint of their adequate fields, because a methodology provides a method to express requirements as a replacement of natural language. So, a methodology must have universality. I believe that "an adequate field fit to the methodology" is not a correct usage of a term of methodology.**

Prof. Mejri **That's an exciting idea. I agree with Mr. Negoro, saying that a methodology is one form of expression of a problem.**

Pfor. Fujita Let me then proceed to the next question. Since Prof. Mejri has just started to learn Lyee, it may be a little tough to ask you to give us a clear vision on Lyee. But what do you think of the position or category that Lyee falls into in computer science? As a computer scientist, do you have some words to say?

Prof. Mejri **At this moment, I want to refrain from giving you a definite perception that Lyee is categorized in such and such an area in comparison with other methods. But if I am forced to say something, I could safely say that Lyee is close to declarative languages. Certain parts of Lyee surely resemble declarative languages. But some other parts of Lyee look like parallel processing. Lyee's way of definition is close to the object oriented. Just by focusing on some specific part of Lyee, there are many things to say, but these comments do not cover a full picture of Lyee. I guess that Lyee requires a totally new field of its own.**

Someone may claim that judging from the way of

defining words and some classifications existing in Lyee, the Lyee method itself is quite similar to the object oriented. Taking a close look at them, however, they are totally different. The same thing can be said for declarative languages. It is true that declarative languages have iteration functions, but with Lyee, the rules of the seven boxes in the predicate structure determine iteration. This is unique to Lyee. A universal structure is embedded in Lyee. Other languages or methods do not work in this way. People are likely to compare Lyee with other methods in order to have a better understanding of Lyee. But what I would like to do in the project is in fact opposite, that is, to look into other methods by using Lyee.

Prof. Hamid The next question is related to the International Scientific Joint Research Project, would you tell us how you would possibly join the project?

Prof. Mejri **Looking at various tools of Lyee for the last few days, I had several ideas popping out in mind. I am particularly interested in legacy conversion. Currently, a certain class of a specific language can be converted into a certain class of another specific language. But I believe it is possible to make a conversion between large classes or any arbitrary languages. One operating system can be transformed into another operating system.**

I am also thinking of generation of parallel programs from Lyee structured programs. This might be considered as a subject. What this aims to do as an idea is eventually the same as what LyeeALL does, but it is different in that the produced outcome is parallel programs. These parallel programs can be executed when they are connected to various networks. That is to say, they are in parallel with LyeeALL. If this can be realized, program generation will be even faster.

I am also thinking of optimization of programs

generated by LyeeALL, particularly in terms of memories and run-time. This part of optimization can be connected with LyeeALL to produce the most optimized programs. This can also be expressed with specific programs for optimization, being independent of LyeeALL I guess either way works out. If independent programs are to be made for optimization, LyeeALL-generated programs will be input for the optimization programs. The outcome of this will be the optimized programs. Optimization cannot be underestimated, because optimized programs will reduce both memory consumption and run-time.

As for verification of the LyeeALL-generated programs, I have some comments. Needless to say, those programs reflect many policies of the customer. I believe it is a very good idea to make a tool which verifies whether those policies expressed in the programs would accord with requirements or not. In other words, the customer wants to express their policies through the requirements so that they want to know the LyeeALL-generated programs could represent their policies, say in security programs or in business programs. For example, a banking institution has a variety of policies such as the one that a balance between output and input should be stricken. That could not be neglected.

I also have an idea of making a tool for checking legacy conversion. The customer wants to know the converted programs could produce the same results as their original programs. If we have a tool to automatically prove that converted programs perform the sama way as the original one, that would increase confidence of the customer in Lyee's legacy conversion. With this tool, the customer does not have to examine for themselves. In Tama Software Factory, I asked workers what the customer did when the legacy conversion was finished.

The customer came to the factory to check whether converted programs could perform the same as their original ones or they took the converted programs to their office and check them. And I asked the workers, "If there were an automatic checking tool, wouldn't it be convenient?" They all said, "Yes."

Another point is related to requirement definition. Currently, the customer gives attributes of words to the Institute as part of their requirements. But those attributes are only associated with computer, not with their ideas. So, this part of work could be simplified as well with a tool. Evolving this part furthermore, it could be completely replaced by a tool. That is, this part of a human work would be removed from the development process. Even though a complete automation is not achievable, I think the highest level of simplification will be anyway possible.

These are the points I am thinking right now. But when I start to actually use LyeeALL, I am sure other interesting thought will come out.