Automation Research in Computer Science

Aragats Amirkhanyan
Hasso Plattner Institute
aragats.amirkhanyan@hpi.de

The main problem for researchers is the lack of data or lack of the necessary environment, or both. Researchers are forced to start from the preparation phase of research. The report consider the concept of the project, which could solve some problems of preparation stage for researches such as deployment research environment, data generation, applying common scenarios that can be reused and extended. The concept project contains 4 systems. Each system can be used independently or in combination with others to meet all requirements. The project does not take on the task of helping to automate all processes of researches, but only preparation stages and scenarios for base researches in Computer Scenes.

1. Introduction

Many research projects come from the industrial companies. It is seems that if the company is interested in success result of the research, the should provide all necessary information, data, research environments. But they do not it. And obviously, the reason has a secure aspect. Most of researchers have to spend a big part of their time for preparing or generate data, deploy research environments and so on. The lack of the data is a big problem of the academic world. The preparation stage has quite big overhead of the research work, that could be overcome in some cases. It is not a secret that many researchers sometimes do almost the same researches that require the same data, the same environments. So it means they almost the same preparation stage. They do some common base scenarios. And every time they have to invent new bicycle and in most cases it is just a waste of time. It would be nice to have some tool for overcoming this overhead, to automatize the common research processes and make research easier and give opportunity to concentrate on the research purposes.

In this report we mostly speak about the problems of preparation stage for researches in Computer. Of course, we do not try to cover the all researches in Computer Science, but we try to cover researches requiring some research environment as a computer network with hosts, routers, firewalls, installed different softwares and generating data for the research environments based on some pre-defined common scenarios such as user login, logout events, installation, connection between host, running service and so on.

The report includes the overview, thoughts, ideas about the prototype application that has the main goal is to simplify the preparation stage of the research and make possible to overcome the research overhead. Despite the fact that the report considers the project for researches it could be used by others developers, administrators and any users.

2. Base Structure

We suggest the concept of the project consists of four independent projects. Each project could be used independent and will be useful for researches, users, administrators, developers. Project could be used together or in any combination to solve different research problems. We call the project Security Lab Generator. It contains four projects (systems):

- Network Designer Service (ND)
- Provision Language Translator (PLT)
- Provision System (PS)
- Network Security Analyzer (NSA)

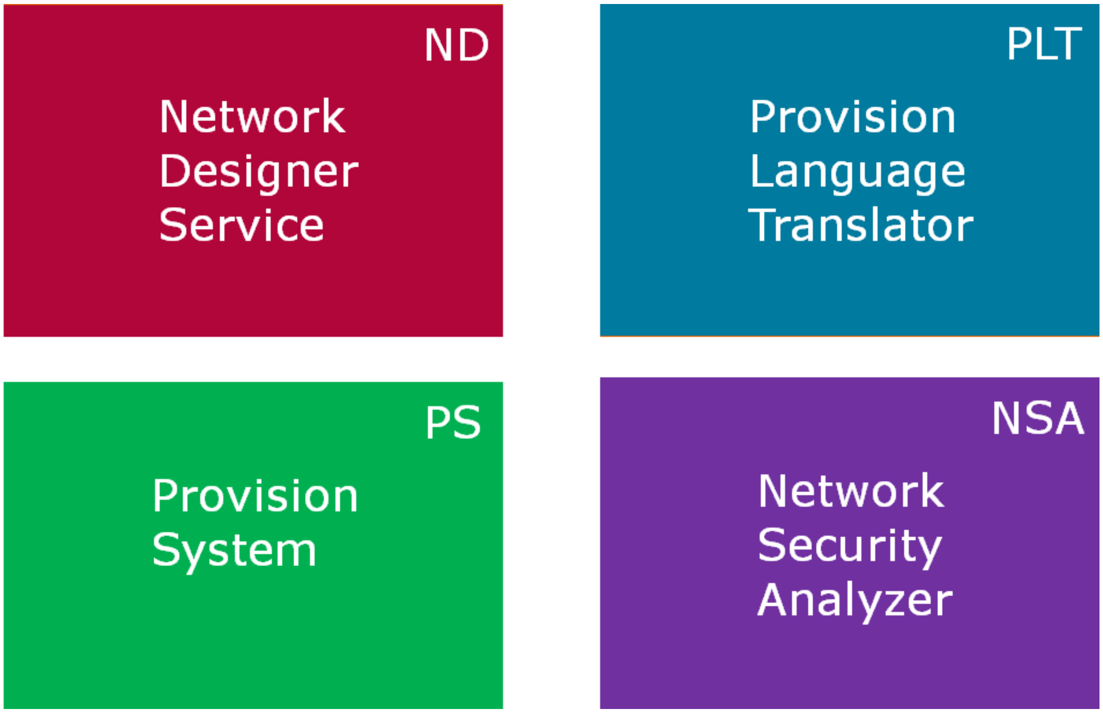


Figure 1: *Security Lab Generator*

As it was mentioned above the concept supposed that all systems can be independently used. It supposed that must be some specified application programming interface (API) for communication between each other. Also it should provide the web user interface for convenient work. It is important to provide flexibility not only in communication between system, but also between modules inside the systems. It means that most of system must be able to extended by additional modules, functionality written by third-party developers. The main idea is combining any modules, systems, reusing some components to able to build needed custom project for solving different tasks.

We just say that everything must be specified, but we do not talk how. It will be done on the stage of developing the architecture and it is out of scope of the report. So now it does not matter which protocol to use REST, SOAP or other and how to integrate the modules.

3. Network Designer Service

The purpose of the Network Designer Service is to provide easy way to design the network including specifying networks, hosts, connections, software, user accounts. The system should provide the flexibility providing by IaaS and simplicity providing by PaaS. It the first part of the big ecosystem. The system should provide the capability to export designed network as structured data. For example XML, JSON or other. It could be used independently by system administrator to design the network and collaboration. And also it can be used in combination with others system to perform the whole life cycle flow. As we mentioned in the previous subsection we want to create project for automation of research. So it is the first step to reach goal.

On the Figure 2 you can see the network topology of OpenStack. OpenStack is open source IaaS. As many IaaS OpenStack provide functionality to run any instances with any operation systems, creating any network structure. It is very flexible. As result of configuring the network and running the instances we can see the network topology describing the network. It is quite easy to read. But IaaS does not provide simplicity of designing the network.

On the other hand there are PaaS systems such as Jelastic, which admin console is illustrated on Figure 3. PaaS provide simplicity of designing. PaaS usually use already pre-defined network structure for different purposes such as Java applications, PHP, Ruby and so on. In this case administrators mostly do not care about orchestration the network structure, they care about which server to use, which database, needed if the load balancer or not and so on.

The idea to combine two approaches from IaaS and PaaS to make flexible and simple designer to provide ability to create any complex network with specified hosts and specified installed software.

Research questions:

- How to fully describe the network structure as XML, JSON or by other language with ability to use the it as a provision language?
- How to specify the principle of network designer for complex structure by combining flexibility IaaS and simplicity PaaS constructor?

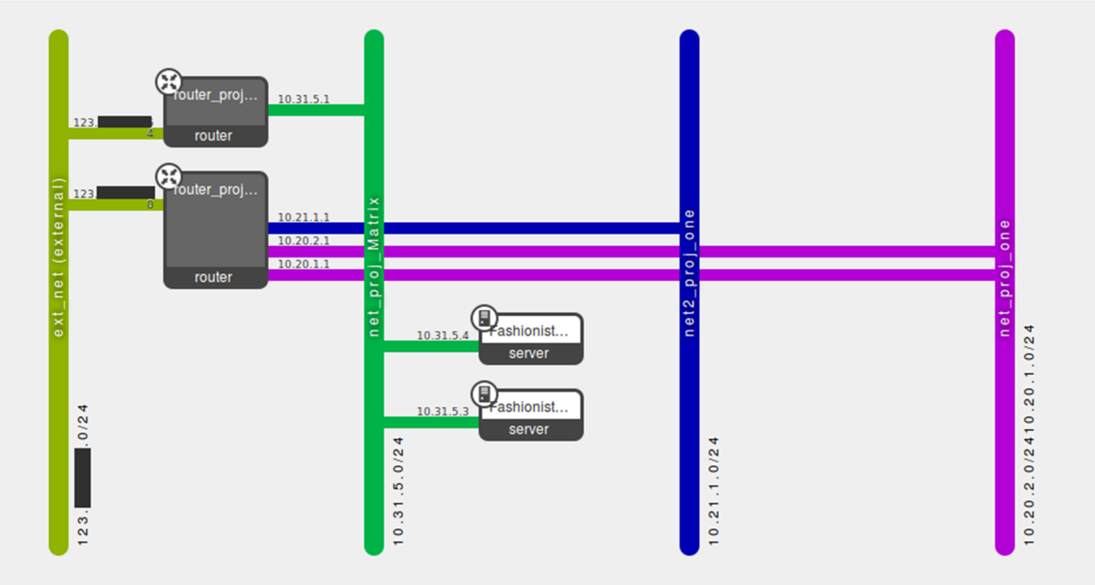
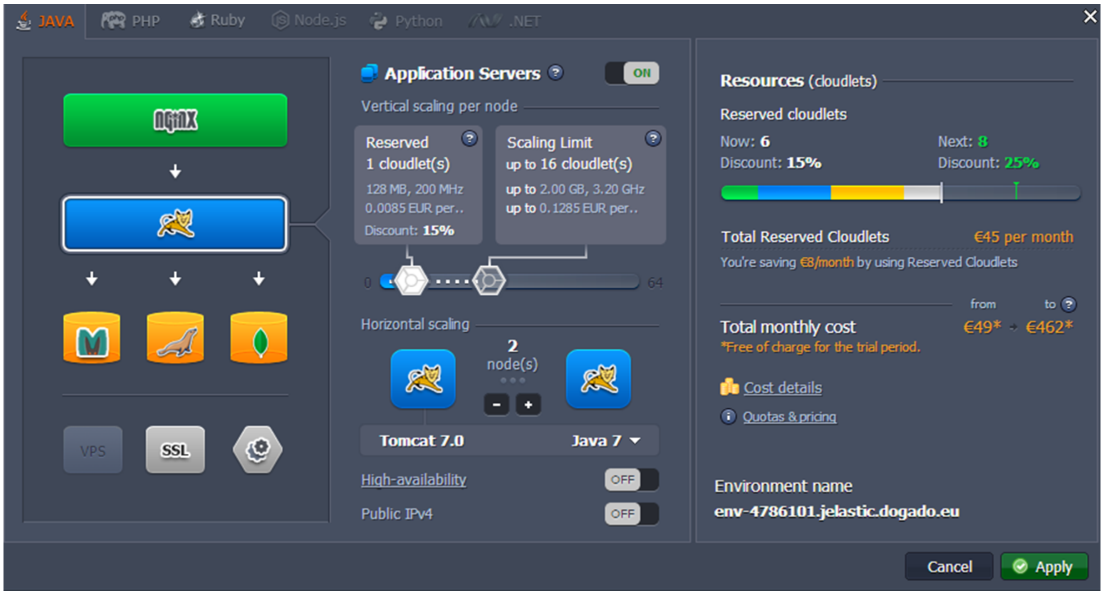


Figure 2: *OpenStack network topology*



Application Servers ON

Vertical scaling per node

Reserved
1 cloudlet(s)
128 MB, 200 MHz
0.0085 EUR per..
Discount: 15%

Scaling Limit
up to 16 cloudlet(s)
up to 2.00 GB, 3.20 GHz
up to 0.1285 EUR per..

0 64

Horizontal scaling

2
node(s)

Tomcat 7.0 Java 7

High-availability OFF

Public IPv4 OFF

Resources (cloudlets)

Reserved cloudlets

Now: 6 Next: 8
Discount: 15% Discount: 25%

Total Reserved Cloudlets €45 per month

You're saving €8/month by using Reserved Cloudlets

from to

Total monthly cost €49* → €462*

*Free of charge for the trial period.

Cost details

Quotas & pricing

Environment name
env-4786101.jelastic.dogado.eu

Figure 3: *Jelastic admin console*

4. Provision Language Translator

Some system provide the capability to describe the environment and deploy it. But each system has its own language. If you do not need to use different provision providers then there is not problem. But it could be that you will need to migrate from one provider to other. For example from Amazon Web Service (AWS) to your own server with VMware EXSi. In this case you have to spend a lot of time to deploy the same environment on your own servers. To automatize the process you would like to use some tools like Vagrant, but it can not resolve all problems of migration, because you still have to write instructions for Vagrant manually. To overcome the problem of migration and provide flexibility in choosing the provider for deploying the environment, we suggest the concept of Provision Language Translator (PLT).

PLT is a translator like Google Translator, but not for nature languages. It is a translator for provision languages. Provision language is a full description of the environment that cloud be used for deploying described environment according to the description. Idea of PLT is to provide so many module as possible for bi-direction translation provision language. The project could be used independently by system administrators for migration network environments from on to other provision provider. But in our case we want to use it in combination with other systems. So we should provide the translation from our network description language to already existing language like Vagrant file or Amazon Cloud Formation or others. So now we have 2 systems ND and PLT. We could design the network, export SLG XML and then translate it into the appropriate provision language.

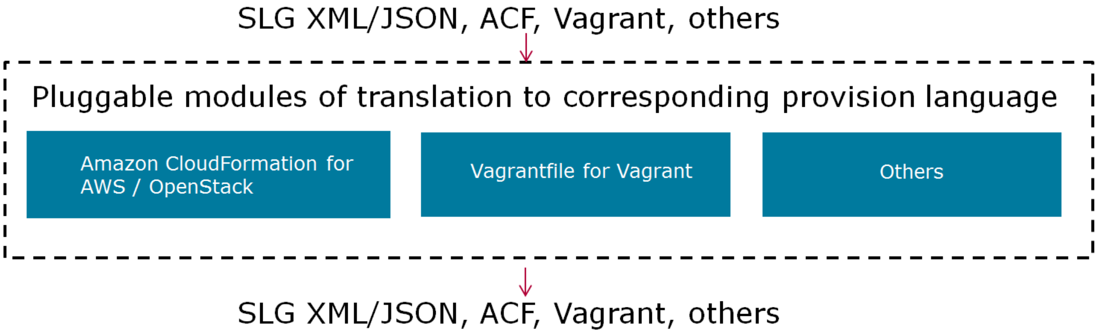

SLG XML/JSON, ACF, Vagrant, others
Pluggable modules of translation to corresponding provision language
Amazon CloudFormation for AWS / OpenStack
Vagrantfile for Vagrant
Others
SLG XML/JSON, ACF, Vagrant, others

Figure 4: *Provision language translator*

Research questions:

- How to convert/translate different provision languages in both direction saving the all structure?

5. Provision System

Provision system is used for deploying the environment based on the description. It can accept different types of provision language and provide provision on different providers: AWS, OpenStack, VMware, VirtualBox and so on. Provision system can directly apply provision language to provider or you can use PLT service to translate input provision language to appropriate for certain provider and then apply it. In combination of three described system we could design the network, convert to appropriate provision language and apply to certain provider.

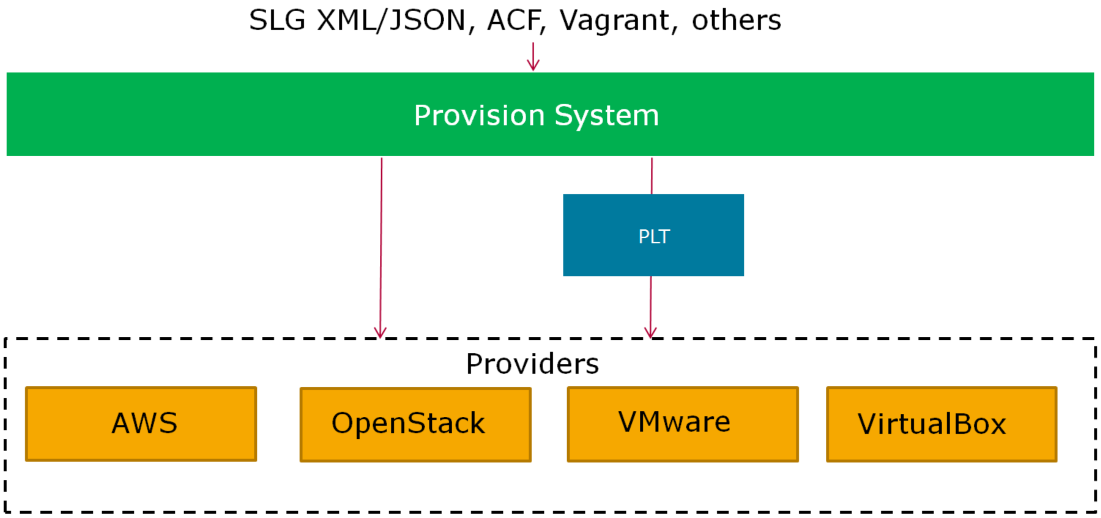


Figure 5: *Provision system*

Research questions:

- How to implement orchestration tool for provision on the different providers?

6. Network Security Analyzer

Network system analyzer is the most interesting part of the project, because it is used for making research, gathering data from host, scanning the hosts, analyze data, making report and applying user scenarios. The system contains two parts: server side and agents.

The server side is wrapper of many security analytic tools such as MulVal. All tools need data to analyze them and create reports, attack graph and show some statistics. So server must know about the whole network structure also gather data from the hosts. The first issue could be solved by passing the SLG XML describing designed network from Network Designer. Already on this step NSA can provide some information such as an attack graph and potential vulnerabilities. But is is not enough. We need information about hosts. For this issue we can use agent.

The NSA agent is client side application. The agent could include different inventory, monitoring, suffering tools to make fully analyzing of the host. Some of possible tools: OVAL SCANNER, NESSUSS and others !!!!. The agent registers in the server and ready to send data to server. The server receive data and depending on the data delegates them to the appropriate analytic tool. The system could get access to other services that could provide useful information for making reports and analyzing data. For example the system could work with Vulnerability

Database (VDB) or RAIMS???!!!

Other very important purpose is creating custom user scenarios, send them to agent and agent apply them. Firstly the concept implies that users can create some description of user scenarios which must be performed on the host side. User scenarios could be some activities such as login,logout, run service, install software, open connections, create files and so on. The NSA must provide the mechanism of simple way of implementation any user scenarios for any operation system. The scenarios must be packed as cookbook (analogy to cookbooks for Chef). It is implied that we will have the repository with set of pre-defined reusable cookbooks. Any user can create his own cookbook, reuse some repository, include his cookbook to repository and combine any cookbooks.

The server takes cookbooks with user scenarios and apply them to host by agents that must be installed on the hosts. As it is just concept we do not distinguish agents by purposed, but we can expect that functionality of gathering data and applying cookbooks could be distributed into different application. But any way both of them are agents and they are installed on the hosts.

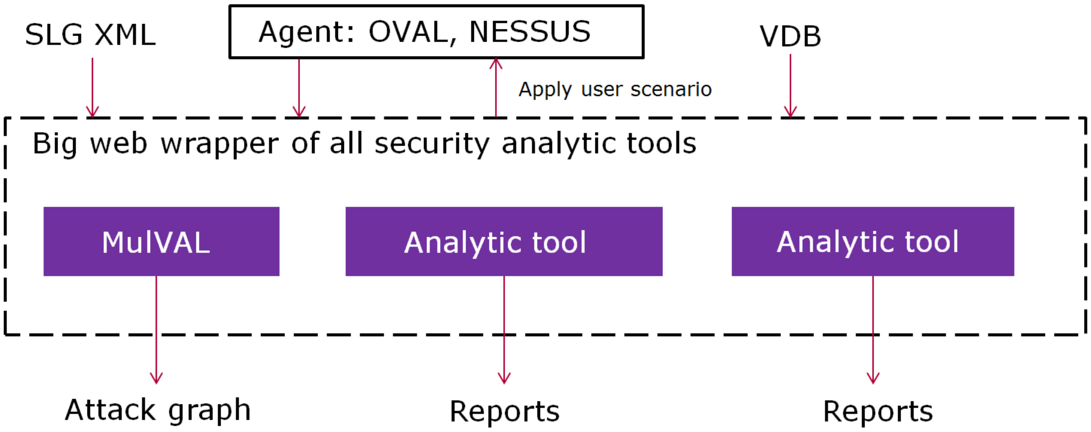


Figure 6: *Network Security Analyzer*

Research questions:

- How to collect all information about the host by special agent wrappers different monitoring tools?
- How to normalize collected data to use them for analyzing?
- How to analyze data?
- How to create reports according collected data and analyzed data?
- How to applying scenarios to hosts.
- How you describe the user scenarios to be able to reuse them and combine with others?

7. Conclusion

To conclude let's remember the main points. In this report we mentioned some problems of researchers that make overhead of the research. The main overhead is because of the preparation stage including creation the research network environment, configuration the network and hosts, installation software, generating data, simulation some typical events. In the report we suggest the concept of the project that can probably solve some of the problems and overcome in some cases the research overhead. We suggest module structure of the project that implies that the project consists of several independent system. And every system can be used independently by administrators, users, researchers or in combination with others.

To prove the idea of concept we gave some usecases of using every system and the whole project. As it is research project we consider possible research questions and ares for researching during the developing the system as a prove of the concept.

8. Methods

Maecenas sed ultricies felis. Sed imperdiet dictum arcu a egestas.

- Donec dolor arcu, rutrum id molestie in, viverra sed diam
- Curabitur feugiat
- turpis sed auctor facilisis
- arcu eros accumsan lorem, at posuere mi diam sit amet tortor
- Fusce fermentum, mi sit amet euismod rutrum
- sem lorem molestie diam, iaculis aliquet sapien tortor non nisi
- Pellentesque bibendum pretium aliquet

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

9. Results

Table 1: *Example table*

Name		
First name	Last Name	Grade
John	Doe	7.5
Richard	Miles	2

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

$$e = mc^2 \tag{1}$$

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

10. Discussion

10.1. Subsection One

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

10.2. Subsection Two

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

References

[1] I. Mizuuchi, R. Tajima, T. Yoshikai, D. Sato, K. Nagashima, M. Inaba, Y. Kuniyoshi, and H. Inoue, “The design and control of the flexible spine of a fully tendon-driven humanoid “Kenta",” in *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, Lausanne, Switzerland, 2002, pp. 2527–2532.

[2] I. Mizuuchi, H. Waita, Y. Nakanishi, T. Yoshikai, M. Inaha, and H. Inoue, “A musculo-skeletal robot leg capable of adding or rearranging the muscles,” in *21th Annual Conference of the Robotics Society of Japan*. Robotics Society of Japan, Tokyo, Japan, 2003, presentation number: 1C29.

[3] I. Mizuuchi, T. Yoshiaki, Y. Nakanishi, and M. Inaba, “A reinforceable-muscle flexible-spine humanoid “Kenji",” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2005, pp. 692–697.

[4] I. Mizuuchi, Y. Nakanishi, Y. Sodeyama, Y. Namiki, T. Nishino, N. Muramatsu, J. Urata, K. Hongo, T. Yoshikai, and M. Inaba, “An Advanced Musculoskeletal Humanoid Kojiro,” in *Proceedings of the 2007 IEEE-RAS International Conference on Humanoid Robotics*, 2007, pp. 101–106.

[5] Y. Nakanishi, Y. Namiki, K. Hongo, J. Urata, I. Mizuuchi, and M. Inaba, “Design of the musculoskeletal trunk and realization of powerful motions using spines,” in *Proceedings of the 2007 IEEE-RAS International Conference on Humanoid Robotics*, 2007, http://planning.cs.cmu.edu/humanoids07/p/85.pdf.

[6] C. Ott, O. Eiberger, W. Friedl, B. Bauml, U. Hillenbrand, C. Borst, A. Albu-Schaffer, B. Brunner, H. Hischmuller, S. Kielhofer, R. Konietschke, M. Suppa, T. Wimbock, F. Zacharias, and G. Hirzinger, “A humanoid two-arm system for dexterous manipulation,” in *Proceedings of the 2006 IEEE-RAS International Conference on Humanoid Robotics*, 2006, pp. 276–283.

[7] J. Or, “A control system for a flexible spine belly dancing humanoid,” *Artificial Life*, vol. 12, no. 1, pp. 63–87, 2006.

[8] J. Or and A. Takanishi, “From lamprey to humanoid: The design and control of a flexible spine belly dancing humanoid robot with inspiration from biology,” *International Journal of Humanoid Robotics*, pp. 81–104, 2005.

[9] ——, “The effect of an emotional belly dancing robot on human perceptions.” *International Journal of Humanoid Robotics*, vol. 4, no. 1, pp. 21–48, 2007.

[10] J. Or, “The development of emotional flexible spine humanoid robots,” in *Affective Computing, Emotion Expression, Synthesis and Recognition*, J. Or, Ed. Advanced Robotics Systems, 2008.