Report Title

Subtitle as needed

Kishan Shah

Master of Software Engineering *(Student)*

Concordia University

Montreal, Canada

shah.kishan0007@yahoo.co.uk

Sohan Argulwar

Master of Applied Computer Science *(Student)*

Concordia University

Montreal, Canada

sohanargulwar@gmail.com

Nirav Bhut

Master of Software Engineering *(Student)*

Concordia University

Montreal, Canada

nirav.bhut1210@gmail.com

Avi Modi

Master of Software Engineering *(Student)*

Concordia University

Montreal, Canada

modiavi91@gmail.com

Tirth Patel

Master of Software Engineering *(Student)*

Concordia University

Montreal, Canada

tirth\_it@yahoo.com

Mohit Pujara

Master of Applied Computer Science *(Student)*

Concordia University

Montreal, Canada

mohitpujara.01@gmail.com

Chilat Shah

Master of Software Engineering *(Student)*

Concordia University

Montreal, Canada

chilatshah1@gmail.com

Hirenkumar R. Tarsadiya

Master of Applied Computer Science *(Student)*

Concordia University

Montreal, Canada

hrtarsadiya647@gmail.com

*Abstract —* The GIPSY system provides a framework for a distributed multi-tier demand-driven evaluation of heterogeneous programs, in which certain tiers can generate demands, while others can respond to demands to work on them. They are connected through a virtual network that can be flexibly configured at run-time. Although the demand generator components were originally designed specifically for the educative (demand-driven) evaluation of lucid intensional programs, the GIPSY's run-time's flexible framework design enables it to perform the execution of various kinds of programs that can be evaluated using the demand-driven computational model. Management of the GISPY networks has become a tedious (although scripted) task that required a manual command-line console, which does not scale for large experiments. Therefore a new component has been designed and developed to allow users to represent, visualize, and interactively create, configure and seamlessly manage such a network as a graph.

Keywords— Demand driven distribution framework, multi dimensional context, hybrid language, run-time system

# Introduction

Give a short description of your study. More importantly, describe the motivation for your study.

# Background

GIPSY bridges gap between intensional and imperative programming languages. Compiler like JLUCID, Objective LUCID and General Imperative promoted the development of GIPSY type system. Overall GIPSY system provides static and dynamic type check for intensional and hybrid programs ***[T].***

Object Oriented Intensional Programming (OO-IP) is a new hybrid language which is a combination of Object Oriented (JAVA) and Intensional Programming (Lucid) languages. The main need to develop hybrid system is combine the power and concrete applicability of intensional programming and the broad acceptability of Object Oriented language. GIPSY provides support for this type of hybrid languages with great flexibility and adaptability which is lacking in the currently used GLU system***[C].***

This framework approach of GIPSY helps developing compiler components of intensional programming languages to execute on language independent run – time system. The data type in java is explicit while Lucid programming language uses implicit data types. So GIPSY provides the mechanism of automatic type casting from imperative to intensional and vice versa at runtime***[S].***

The main goal of GIPSY is to provide generality, adaptability and efficiency. It is divided into 3 subsystems, the General Intensional Programming Language Compiler (GIPC), the General Education Engine (GEE) and the Intensional Run-time Programming Environment (RIPE). Here GIPL translates the LUCID code into Intensional Data dependency Structure (IPS) and Intensional Communication procedures (ICP) for data communication, GEE computes the remote procedure call and stores it in the warehouse and hence it helps in reducing the overhead generated by the procedure call, the RIPE module enables us to visualize the dataflow diagram which in turn enables us to dynamically inspect warehouse values and change i/o channels of the program ***[A].***

Earlier it was not possible to assign a context to a variable, pass as a parameter, return as a result of a function and that was necessary for integrating LUCID variants to GIPSY system. To resolve this issue Lucx parser has been developed and by that means we got efficiency of execution much higher than the classical translation methods proposed previously ***[H].***

GIPSY is a demand driven educational model and it creates chain of demands while translating any intentional programming language into GIPL. JINI and JMS are two JAVA distributed middleware used in the implementation of demand migration framework. By doing so we are refactoring their parent DMF ***[K].***

To advance software engineering design and implementation of the multi-tier run-time system for GIPSY we decided to integrate four local and distributed computation prototypes which are multi-threaded, RMI, Jini and JMS together by applying the abstract factory, factory method and strategy design patterns***[M].***

Organization and management of GIPSY networks were very complex and time consuming, which used to require manual handling using command line console. So new interactive graph based assistant component has been developed that allow users to picture, represent and easily create, configure and manage those networks virtually as a graph. It also allow users to control the all related network parameters and the interconnection among nodes at run time. The main aim or objective of developing is graph-based graphical user interface is to increase the usability of the running system and allowing the user to have a control over the network with very less manual intervention ***[N].***

## Sub-section 1

Discuss papers related to your study. Break into sub-sections if necessary.

## Sub-section 2

## Summary

##### References

[MT]

Serguei A. Mokhov, Joey Paquet, and Mourad Debbabi. The use of NLP techniques in static code analysis to detect weaknesses and vulnerabilities. In Maria Sokolova and Peter van Beek, editors, Proceedings of Canadian Conference on AI'14, volume 8436 of LNAI, pages 326{332. Springer, May 2014. ISBN 978-3-319-06483-3. doi: 10.1007/978-3-319-06483-3 33. Short paper.

[MA]

Serguei A. Mokhov. Choosing best algorithm combinations for speech processing tasks in machine learning using MARF. In Sabine Bergler, editor, Proceedings of the 21st Canadian AI'08, LNAI 5032, pages 216{221, Berlin Heidelberg, May 2008. Springer-Verlag. doi: 10.1007/978-3-540-68825-9 21.

[MM]

Serguei A. Mokhov, Miao Song, and Ching Y. Suen. Writer identi\_cation using inexpensive signal processing techniques. In Tarek Sobh and Khaled Elleithy, editors, Innovations in Computing Sciences and Software Engineering; Proceedings of CISSE'09, pages 437{441. Springer, December 2009. doi: 10.1007/978-90-481-9112-3 74. ISBN: 978-90-481-9111-6, online at: http: //arxiv.org/abs/0912.5502.

[MN]

Serguei A. Mokhov. Combining and comparing multiple algorithms for better learning and classi\_cation: A case study of MARF. In Suraiya Jabin, editor, Robot Learning, chapter 2, pages 17{42. InTech, August 2010. doi: 10.5772/10248. ISBN: 978-953-307-104-6, online at http://www.intechopen.com/download/pdf/pdfs\_id/12131.

[MS]

Serguei A. Mokhov. Towards syntax and semantics of hierarchical contexts in multimedia processing applications using MARFL. In Proceedings of the 32nd Annual IEEE International Computer Soft-ware and Applications Conference (COMPSAC), pages 1288{1294, Turku, Finland, July 2008. IEEE Computer Society. doi: 10.1109/COMPSAC.2008.206.

[MC]

Serguei A. Mokhov. The use of machine learning with signal- and NLP processing of source code to \_ngerprint, detect, and classify vulnerabilities and weaknesses with MARFCAT. Technical Report NIST SP 500-283, NIST, October 2011. Report: <http://www.nist.gov/> manuscript-publication-search.cfm?pub\_id=909407, online e-print at <http://arxiv.org/abs/> 1010.2511.

[MK]

Serguei A. Mokhov. L'approche MARF \_a DEFT 2010: A MARF approach to DEFT 2010. In Proceedings of the 6th DEFT Workshop (DEFT'10), pages 35{49. LIMSI / ATALA, July 2010. DEFT 2010 Workshop at TALN 2010; online at http://deft.limsi.fr/actes/2010/pdf/2\_clac.pdf.

[MH]

Serguei A. Mokhov. Study of best algorithm combinations for speech processing tasks in machine learning using median vs. mean clusters in MARF. In Bipin C. Desai, editor, Proceedings of C3S2E'08, pages 29{43, Montreal, Quebec, Canada, May 2008. ACM. ISBN 978-1-60558-101-9. doi: 10.1145/1370256.1370262.

[GT] Serguei A. Mokhov, Joey Paquet, and Xin Tong. A type system for hybrid intensional-imperative programming support in GIPSY. In Proceedings of the Canadian Conference on Computer Science and Software Engineering (C3S2E’09), pages 101–107, New York, NY, USA, May 2009. ACM. ISBN978-1-60558-401-0. doi: 10.1145/1557626.1557642

[GA] Joey Paquet and Peter G. Kropf. The GIPSY architecture. In Peter G. Kropf, Gilbert Babin, John Plaice, and Herwig Unger, editors, Proceedings of Distributed Computing on the Web, volume 1830 of Lecture Notes in Computer Science, pages 144–153. Springer Berlin Heidelberg, 2000. doi:10.1007/3-540-45111-0 17.

[GM] Bin Han, Serguei A. Mokhov, and Joey Paquet. Advances in the design and implementation of a multi-tier architecture in the GIPSY environment with Java. In Proceedings of the 8th IEEE/ACIS International Conference on Software Engineering Research, Management and Applications (SERA2010), pages 259–266. IEEE Computer Society, May 2010. ISBN 978-0-7695-4075-7. doi: 10.1109/SERA.2010.40. Online at http://arxiv.org/abs/0906.4837

[GN] Sleiman Rabah, Serguei A. Mokhov, and Joey Paquet. An interactive graph-based automation assistant: A case study to manage the GIPSY’s distributed multi-tier run-time system. In Ching Y.Suen, Amir Aghdam, Minyi Guo, Jiman Hong, and Esmaeil Nadimi, editors, Proceedings of the ACM Research in Adaptive and Convergent Systems (RACS 2013), pages 387–394, New York, NY,USA,October 2011–2013. ACM. ISBN 978-1-4503-2348-2. doi: 10.1145/2513228.2513286. Pre-print: http://arxiv.org/abs/1212.4123

[GS] Serguei A. Mokhov and Joey Paquet. A type system for higher-order intensional logic support for variable bindings in hybrid intensional-imperative programs in GIPSY. In Tokuro Matsuo,Naohiro Ishii, and Roger Lee, editors, 9th IEEE/ACIS International Conference on Computerand Information Science, IEEE/ACIS ICIS 2010, pages 921–928. IEEE Computer Society, May2010. ISBN 978-0-7695-4147-1. doi: 10.1109/ICIS.2010.156. Presented at SERA 2010; pre-print at http://arxiv.org/abs/0906.3919

[GC] Aihua Wu, Joey Paquet, and Serguei A. Mokhov. Object-oriented intensional programming: Intensional Java/Lucid classes. In Proceedings of the 8th IEEE/ACIS International Conference on Software Engineering Research, Management and Applications (SERA 2010), pages 158–167. IEEE Computer Society, May 2010. ISBN 978-0-7695-4075-7. doi: 10.1109/SERA.2010.29. Pre-print at: http://arxiv.org/abs/0909.0764

[GK] Yi Ji, Serguei A. Mokhov, and Joey Paquet. Unifying and refactoring DMF to support concurrent Jini and JMS DMS in GIPSY. In Bipin C. Desai, Sudhir P. Mudur, and Emil I. Vassev, editors,Proceedings of the Fifth International C\* Conference on Computer Science and Software Engineering (C3S2E’12), pages 36–44, New York, NY, USA, June 2010–2013. ACM. ISBN 978-1-4503-1084-0.doi: 10.1145/2347583.2347588. Online e-print http://arxiv.org/abs/1012.2860

[GH] Joey Paquet, Serguei A. Mokhov, and Xin Tong. Design and implementation of context calculus in the GIPSY environment. In Proceedings of the 32nd Annual IEEE International Computer Software and Applications Conference (COMPSAC), pages 1278–1283. IEEE Computer Society, July 2008.doi: 10.1109/COMPSAC.2008.200.