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

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

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

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

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

1. G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*
2. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
3. I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
4. K. Elissa, “Title of paper if known,” unpublished.
5. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
6. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
7. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.