**Towards Syntax and semantics of Hierarchical Contexts in Multimedia Processing Applications using MARFL**

MARF (Modular Audio Recognition Framework) is a collection of voice/text/speech and Natural Language Processing (NLP) algorithms written in Java and arranged in modular and extensible framework which provides addition of new algorithms. This research paper focus on syntax and semantics for MARF Language. Scripting in required application is difficult task by providing context of all parameters. The context expressions provides scripting MARF based applications as context aware. To make the syntax simpler the overloaded context operators to accept various types of arguments and return types @ and # which are taken from Generic Intensional Programming Language (GIPL) helps to achieve this task. MARF plays and important role in the field of Image Processing and Pattern Recognition. It provides APIs in Java with implementation of unsupervised learning. For example, if user misses any dimension to provide in scripting MARFL it will consider the default value. The paper illustrates the practical application of MARFL as SpeakerIdentApp. The definations of @ and # are overridden to introduce the concept of dot operator for object membership.

References:

<http://marf.sourceforge.net/>

http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4591767&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs\_all.jsp%3Farnumber%3D4591767

**A type system for Higher-Order Intensional Logic Support for variable bindings in Hybrid Intensional – Imperative programs in GIPSY**

This research paper is about building a system - General Intensional Programming System (GIPSY) to support intensional programming languages based on intensional logic and their execution model. This framework approach of GIPSY helps developing compiler components of intensional programming languages to execute on language independent run – time system. In, Lucid programming languages data types are implicit i.e. we have to identify the data type by using its result of the expression. GIPSY tries to allow this automatic type conversion when needed to evaluate hybrid expression. This paper present the connection between Java and Lucid. Paper provides the solution for data type casting between Java and other Lucid programming languages. There are some examples stated in paper which describes the situation we GIPSY automatically type cast the data type accordingly when required. One of the example is of void return type in Java which is matched to Boolean true in Lucid. Generally, intensional languages has dynamic data types thus their value is not known at compile time and hybrid languages are generally compile time. The paper then describes the data types like numeric, logical, composite, existential, intersection, union, linear, etc. The paper concludes by stating the implementation of these data types in intensional and hybrid intensional – imperative languages.

References:

http://arxiv.org/abs/0906.3919