Table 1: GROUP MEMBERS

Name	**Reg No**	**Student No**
Nahurira Humphrey	15/U/9205/PS	215016851
Katende Marvin	15/U/5954/EVE	215010864
Gorlyan Isaac	14/U/6365/PS	
Sendaula Brian	14/U/24011/EVE	

A CONCEPT PAPER FOR ANALYSING CONCURRENT DATA TYPES OF CSP.

April 19, 2017

1 INTRODUCTION.

In computer science Communicating Sequential Processes (CSP) is a formal language for describing patterns of interaction in concurrent systems. It is a member of the family of mathematical theories of concurrency known as process algebras, or process calculi based on message passing via channels.CSP was highly influential in the design of the Occam programming language and also influenced the design of programming languages such as Limbo,RaftLib,Go and Crystal CSP provides two classes of data types of primitives in its process algebra Events. These represent communications or interactions. They are assumed to be indivisible and instantaneous. They may be atomic names (e.g. on, off) compound names (e.g. vave.open, valve.close) or input events (e.g. mouse? xy, screen! bitmap). Primitive processes. Primitive processes represent fundamental behaviours: examples include STOP (the process that communicates nothing, also called deadlock), and SKIP (which represents successful termination).

2 BACKGROUND OF THE PROBLEM.

The version of CSP presented in Hoares original 1978 paper was essentially a concurrent programming language rather than a process calcus.it had a substantially different syntax than later versions of CSP, did not possess mathematically defined semantics, and was unable to represent unbounded nondeterminism.Programs in the original CSP were written as a parallel composition of

a fixed number of sequential processes communicating with each other strictly through synchronous message-passing. In contrast to later versions of CSP,each process was assigned an explicit name and the source or destination of a message was defined by specifying the name of the intended sending or receiving process.

3 PROBLEM STATEMENT.

y An early and important application of CSP was its use for specification and verification of elements of the INMOS T9000 Transputer, a complex superscalar pipelined processor designed to support large multiprocessing.CSP was employed in verifying the correctness of both the processor pipeline and the Virtual Channel Processor which managed off-chip communications for the processor.

4 OBJECTIVES.

4.1 General Objective.

The application of CSP to software design is usually focused on dependable and safety critical systems.Forexample the Bremen institute for safe systems and Daimler-Benz Aerospace modeled a fault management system and avionics interface(consisting of some 23,0000 lines of code) intended for use on the international space station in CSP and analyzed the model to confirm that their design was free of deadlock and livelock.The modeling and analysis process was able to uncover a number of errors that would have been difficult to detect using testing alone.Similarly,Praxis High Integrity Systems applied CSP modeling and analysis during the development of software(approximately 100,000 lines of code) for secure smart-card Certification Authority to verify that their design was secure and free of deadlock. Praxis claims that the system has a much lower detect rate than comparable systems.

4.2 Other objectives include;

1. To analyze any type of data types those are occurring at the same time.
2. To increase on the speed data types are analyzed for a given period of time.
3. To maintain the base in which data types are analyzed.
4. To apply constraints in a way data types are meant to be grouped to increase efficiency.
5. To validate data types analysis basing on the requirements available.

5 METHODOLOGY.

As its name suggests ,CSP allows the description of systems interms of component process that operate independently ,and interact with each other solely through message-passing communication.However,The Sequential part of the CSP name is now something of a misnomer, since modern CSP allows component processes to be defined both as sequential processes, and the parallel composition of more primitive processes. The relationships between different processes, and the way each process communicates with its environment, are described using various process algebraic operators or their data types. Using the algebraic approach, quite complex process descriptions can be easily constructed from a few primitive elements.

6 LITERATURE REVIEW.

Over the years a number of tools for analyzing and understanding systems described using CSP have been produced. Early tool implementations used a variety of machine-readable syntaxes for CSP making input files written for different data types incompatible.However,most CSP data types have now standardized on the machine readable dialect of CSP devised by Bryan Scattergood ,sometimes referred to as CSP.The CSP dialect of CSP processes are formally defined operational semantics which includes embedded functional programming language. .It is released by the university of Oxford ,which also released FDR2 in the period 2008-12.

References

[1] Roscoe, A.W (1997).The Theory and Practice of Concurrency.*Prentice Hall ISBN 0-13-674409-5*

[2] INMOS(1995-05-12) Occam 2.1 Reference Manual(PDF) *SGS-THOMSON Microelectronics Ltd .,INMOS document 72 occ 45 03*

[3] Resources about Threaded programming in the Bell Labs CSP style*Retrieved 2010-04-15*

[4] Language Design FAQ: *Why build concurrency on the ideas of CSP?*

[5] Hoare, C.A.R. (1978).Communicating Sequential Processes. *Communications of the ACM 21(8):666-677.doi:10.1145/359576.359585.*

[6] *Brookes, Stephen; Hoare, C.A.R; Roscoe, A.W. (1984).*A theory of Communicating Sequential Processes.Jouranal of ACM.31(3):560-599.doi:10.1145/828.833