



Vector/Signal/ Image Processing Library (VS IPL) Update

Mark A. Richards

Georgia Tech Research Institute

770-528-7758 <mark.richards@gtri.gatech.edu>

May 1999

Original development of the VS IPL standard was supported by DARPA under U.S. Army Contract No. DABT63-96-C-0060.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of DARPA or the U.S. Army.



The VSIPL Forum

- The *VSIPL Forum* is a volunteer organization made up of representatives from industry, government, users, and academia who are working to define an industry standard API for vector, signal, and image processing primitives for embedded real-time signal processing systems.
- **Current Core Members (active contributors to developing the VSIPL standard) are:**
 - **Vendors**
 - *Mercury, SKY, CSPI, Compaq, Cray/SGI, Khoral Research, TI, Alacron, ORINCON*
 - **Universities & Research Organizations**
 - *HRL-LLC, Georgia Tech, MSU, MITRE, MIT/LL, SDSC*
 - **DoD and Government**
 - *DARPA, PEO USW, SPAWAR SSC San Diego*
 - **Users**
 - *LNK, Lockheed-Martin GES, LM/Sanders, Northrop-Grumman, Raytheon*
- **Current Co-Chairs are Randy Janka (GTRI), Jim Lebak (MIT/LL), and Mark Richards (GTRI)**



VSIP L Goals

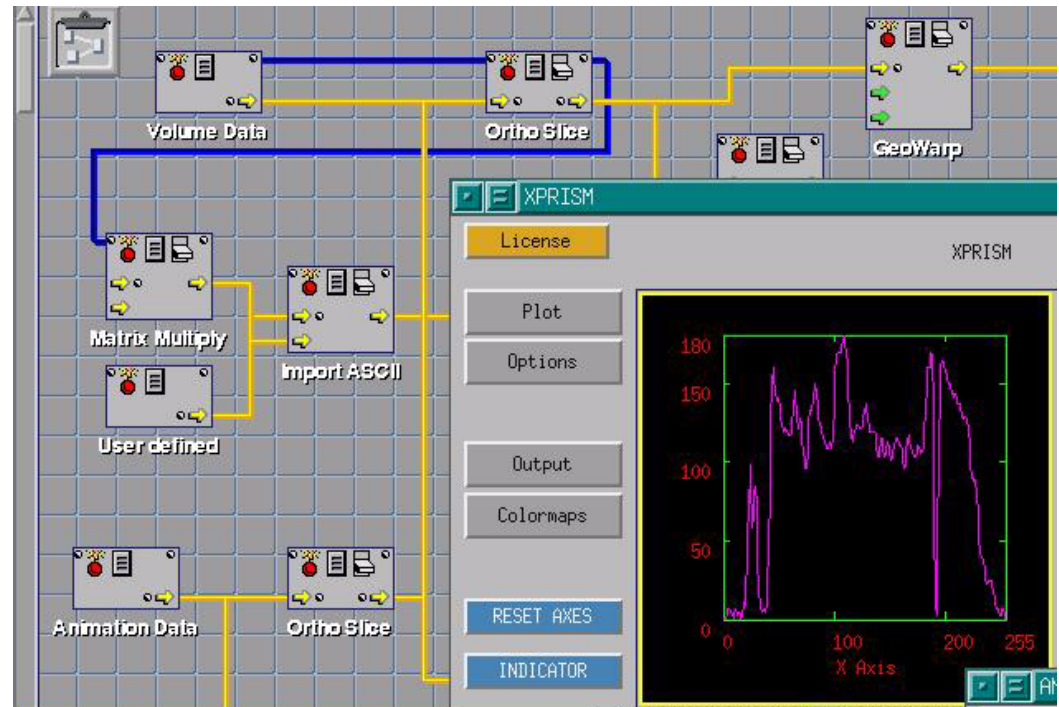
- **Create a widely accepted *and adopted* industry-supported standard API/library for vector/signal/image processing primitives**
- **Foster standardization for sensor software portability**
 - reuse, interoperability, low cost COTS upgrade path, lower life cycle costs, etc.
- **Support migration of applications developed on workstations to embedded systems**
- **Support development of portable middleware environments**



High Level Tools for Embedded Processors

VSIP L facilitates building higher level tools and middleware that can target many platforms

- Possible tools include:
 - Khoros (Khoral Research)
 - Ptolemy (UC Berkeley)
 - Processing Graph Method (PGM) Tool (NRL) and Autocoding Toolset (MCCI)
 - Scalable Programming Environment (SPAWAR SSC)
 - GEDAE (Lockheed-Martin)
 - Rippen (ORINCON)
 - MATLAB and Simulink (The Mathworks)
- Possible Middleware
 - TASP COE
 - Sonar Middleware Standard





Scope of VSIP L API

- **Vector Processing**
 - vector and matrix processing
- **Signal Processing**
 - FFT, filters, support functions
- **Image Processing**
 - not currently defined
- **Linear Algebra**
 - basic processing, decompositions, solvers
- **Support Functions**
 - abstracted memory management functions
 - other assorted support functions for VSIP L



VSIPPL Function Summary

Function Summary

- Support
- Vector/Matrix
- Linear Algebra
- Signal
- Image TBD

Block Objects

Memory Block Create
 Memory Block Bind
 Memory Block Find
 Memory Block Destroy
Vector View Objects
 Create a Vector View
 Destroy Vector View
 Get Vector View Attributes
 Put Vector View Attributes
 Get a Vector Element
 Put a Vector Element
 Find First Vector Index
 User Defined Boolean Vector Binary Function
 User Defined Binary Vector Function
 User Defined Unary Vector Function
 User Defined Vector Generator
 User Defined N-ary Vector Function
 Serial User Defined N-ary Vector Function
 Vector Summation
 Vector Create
 Vector View Clone
 Vector Get Block
 Vector Get Offset
 Vector Get Stride
 Vector Get Length
 Vector Put Block
 Vector Put Offset
 Vector Put Stride
 Vector Put Length
 Create Real Vector View
 Create Imaginary Vector View
 Create Sub-View Vector View
Matrix View Objects
 Create a Matrix View
 Destroy Matrix View
 Get Matrix View Attribute
 Put Matrix View Attribute
 Get a Matrix Element
 Put a Matrix Element
 Matrix Create
 Matrix View Clone
 Matrix Get Block
 Matrix Get Offset
 Matrix Get Stride
 Matrix Get Stride
 Matrix Get Length
 Matrix Get Length

Matrix Put Block
 Matrix Put Offset
 Matrix Put Stride
 Matrix Put Stride
 Matrix Put Length
 Matrix Put Length
 Create Real Matrix View
 Create Imaginary Matrix View
 Create Sub-View Matrix View
 Create Row-View Matrix View
 Create Column-View Matrix View
 Create Diagonal-View Matrix View
 Create Transposed-View Matrix View
 User Defined Boolean Matrix Binary Function
 User Defined Binary Matrix Function
 User Defined Unary Matrix Function
 User Defined Matrix Generator
 User Defined N-ary Matrix Function
 Serial User Defined N-ary Matrix Function
Elementary Math Functions
 Vector/Matrix Arccosine
 Vector/Matrix Arcsine
 Vector/Matrix Arctangent
 Vector/Matrix Arctangent of Two Arguments
 Vector/Matrix Cosine
 Vector/Matrix Cosine
 Vector/Matrix Exponential
 Vector/Matrix Exponential Base 10
 Vector/Matrix Log
 Vector/Matrix Log Base Ten
 Vector/Matrix Sine
 Vector/Matrix Sine
 Vector/Matrix Square Root
 Vector/Matrix Tangent
 Vector/Matrix Tangent
Unary Operations
 Vector/Matrix Magnitude
 Vector/Matrix Argument
 Vector/Matrix Conjugate
 Vector/Matrix Magnitude Squared
 Vector/Matrix Mean Value
 Vector/Matrix Mean Square Value
 Vector/Matrix Negate
 Vector/Matrix Reciprocal
 Vector/Matrix Inverse Square Root
 Vector/Matrix Square
 Vector/Matrix Sum Value

Vector/Matrix Sum of Squares Value
Binary Operations
 Vector/Matrix Add
 Vector/Matrix Divide
 Vector/Matrix Divide Scalar
 Vector Dot Product
 Vector/Matrix Hypotenuse
 Vector/Matrix Conjugate Multiply (Elementwise)
 Vector/Matrix Multiply (Elementwise)
 Vector/Matrix Polar
 Vector/Matrix Rectangular
 Vector/Matrix Scalar Add
 Scalar Vector/Matrix Divide
 Vector/Matrix Scalar Multiply
 Vector/Matrix Subtract
Ternary Operations
 Vector Add and Multiply
 Vector Multiply and Add
 Vector Multiply, Scalar Add
 Vector Multiply and Subtract
 Vector Scalar Add, Vector Multiply
 Vector Subtract and Multiply
 Vector Scalar Multiply, Vector Add
 Vector Scalar Multiply, Scalar Add
Logical Operations
 Vector/Matrix Logical Equal
 Vector/Matrix Logical Greater Than or Equal
 Vector/Matrix Logical Greater Than
 Vector/Matrix Logical Less Than or Equal
 Vector/Matrix Logical Less Than
 Vector/Matrix Logical Not Equal
 Vector/Matrix Some True
 Vector/Matrix All True
Selection Operations
 Vector/Matrix Clip
 Vector/Matrix Index a Boolean
 Vector/Matrix Maximum
 Vector/Matrix Maximum Magnitude
 Vector/Matrix Maximum Magnitude Squared
 Complex Vector/Matrix Max Squared Value
 Vector/Matrix Maximum Magnitude Value
 Vector/Matrix Maximum Value
 Vector/Matrix Minimum
 Vector/Matrix Minimum Magnitude

Complex Vector/Matrix Min Magnitude Squared
 Complex Vector/Matrix Min Squared Value
 Vector/Matrix Minimum Magnitude Value
 Vector/Matrix Minimum Value
 Vector/Matrix Threshold
Bitwise/Boolean Operators
 Vector/Matrix AND
 Vector/Matrix NOT
 Vector/Matrix OR
 Vector/Matrix Exclusive OR
Element Generation
 Vector/Matrix Fill
 Vector Ramp
 Vector/Matrix Random Gaussian
 Vector/Matrix Random Uniform
Manipulation Operations
 Vector/Matrix Complex
 Vector/Matrix Imaginary
 Vector/Matrix Real
 Vector/Matrix Swap
 Vector/Matrix Gather
 Vector/Matrix Scatter
Basic Matrix Functions
 Matrix Transpose Product
 Matrix Hermitian Product
 Matrix Product
 Matrix Conjugate Product
 Generalized Matrix MAC
 Matrix Vector Product
 3 by 3 Matrix Vector Product
 4 by 4 Matrix Vector Product
 Matrix Transpose
 Matrix Conjugate Transpose
 Generalized Matrix Accumulate
 Kronecker Matrix Product
 Outer Vector/Matrix Product
Linear System Solvers
 Matrix LU Decomposition
 Matrix QR Decomposition
 Matrix Cholesky Decomposition
 LU Linear System Solver
 QR Linear System Solver
 Cholesky Linear System Solver



Key VSIP Library Concepts

- **Library-based memory hierarchy performance addressed by**
 - **Opaque Abstract Data Types**
 - **VSIP Admitted (VSIP data) and Released (User data) Memory (blocks)**
 - **Deferred Execution**
 - **Supports vendor optimizations for**
 - *loop fusion*
 - *strip mining*
 - *(vector) chaining*
 - *hand optimized assembly language*
 - *optimized memory hierarchy and multiple address spaces*
 - *system specific services and OS support*
 - *optimal unrolling and software pipelining*
 - *et cetera*
- **Object-based**
 - **blocks, vectors, matrices, images, ...**
 - **Vectors and Matrices are *views* of blocks**

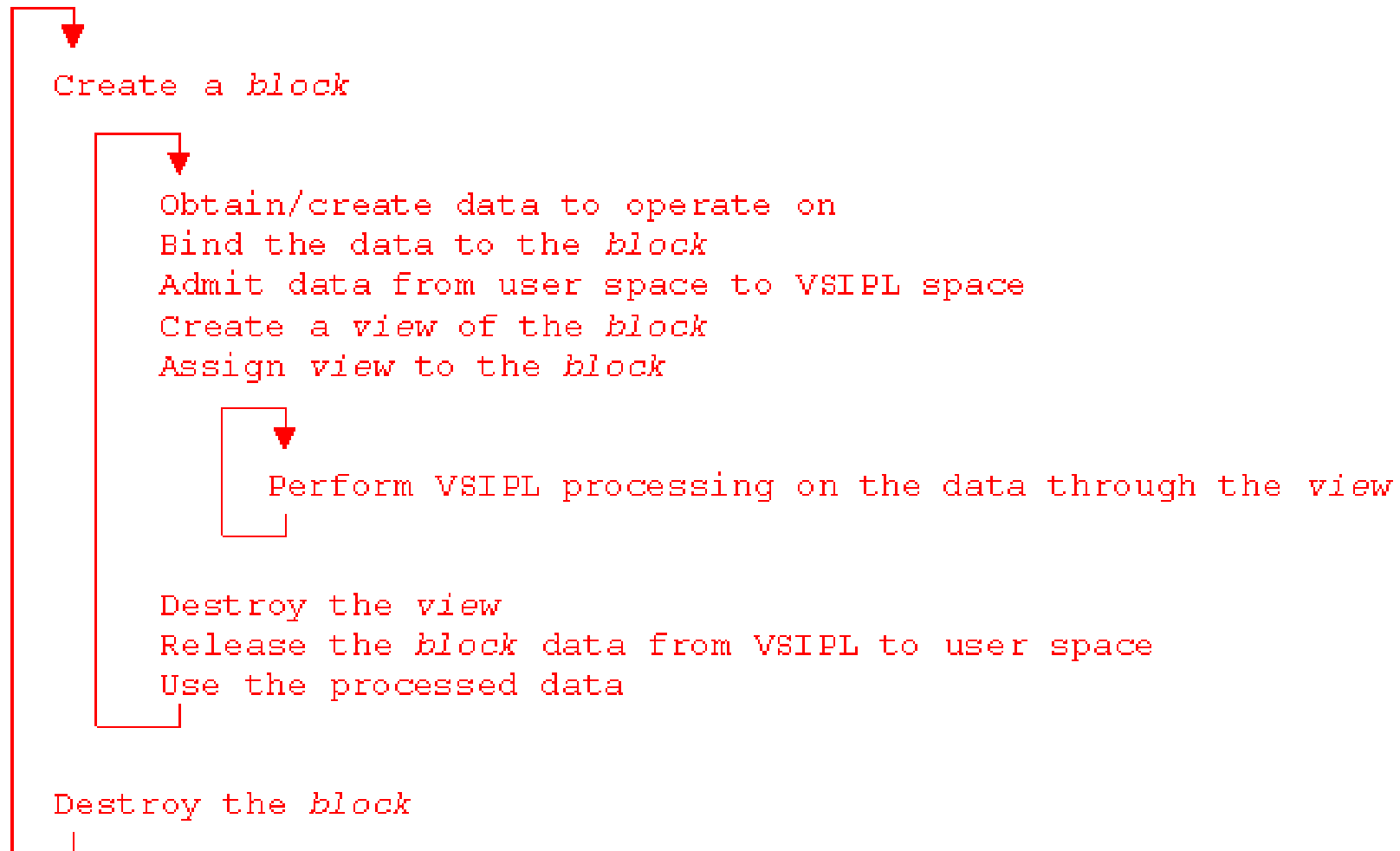


VSIP C Library Properties

- Based On ANSI C
- Vectors, Matrices, and 3-Tensors based *conceptually* on contiguous ANSI C arrays.
 - Matrices can be *row-major* or *column-major* to simplify mixed library support.
- Vectors, Matrices, and 3-Tensors, Real and Complex, are *views* on data *blocks*
 - Multiple views may be bound to a block.
- Library *namespace* is used to minimize runtime data directed dispatching and to minimize linked module size.
- Two modes:
 - Development with extensive checking
 - Production with negligible checking



Basic VSIPL Coding Approach





Data Types

- **Standard Types**
 - float, double, long
 - char, uchar, int, uint, short int, long int, *etc.*
 - boolean, index types, *etc.*
- **Portable Precision Types**
 - minimum number of digits, *.e.g.* `_f6`
 - minimum number of bits, *.e.g.* `_i132`
 - fastest minimum number of bits, *.e.g.* `_if32`
 - portable applications should use these types only
- **Non-Portable Types**
 - exact number of bits, *.e.g.* `_ie32`
 - take advantage of specific hardware features



VSIP Source Code Complexity Comparison

- Based on VSIP as defined 8/97 (“VSIP”)
- See <www.vsipl.org> “examples” page for full tech memo

Environment	Source Lines of Code				Single or Double Precision?	New Functions Assumed?
	Total	Setup	Control	Computation		
MATLAB	11	1	0	10	D	N
<i>C Implementations</i>						
VSIP	52	27	2	23	D	N
Mercury SAL	54	23	3	28	S	N
SKY SKYvec	59	29	3	27	D	Y
CSPI ISSPL	49	19	3	27	S	N
Sonar Middleware Standard	62	28	3	31	S	Y
MATLAB Compiler	174	55	14	105	D	N
<i>C++ Implementations</i>						
MATCOM C++ compiler	33	13	3	17	D	N
VSIP C++ binding	12	4	1	7	D	Y



VSIPL Forum Products

- **Draft Standard API for Vector/Signal Processing**
 - Image processing deferred
- **Single Processor VSIPL API/Library**
 - C Reference Implementation by HRL
 - C “Core Lite” performance mode implementation
- **Portable C Test/Compliance Suite**
- **E-mail Listserver and Web site**
 - Distribution of comments, discussions, drafts, standards and software for public distribution



VSIPL Users and Commercial Products



- **Early Experimenters**

- TASP COE



- San Diego SSC PET Team



- MITRE STAP benchmark conversion

- **Products in Development**



- MPI Software Technology VSI/Pro



- Annapolis MicroSystems WildStar FPGA Boards



- SKY Computers Rome Lab system



- Mercury & CSPI in development



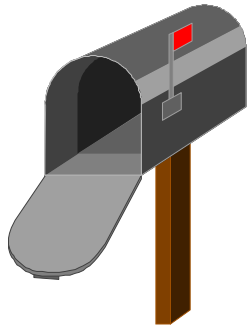
VSIP L Status

- **API Standard**
 - 1.0 specification done except for the final documentation.
 - Completion and adoption of Version 1.0 by VSIP L Forum expected in FY99.
- **Software**
 - **Workstation demonstration libraries available for Core Lite profile**
 - *developed by SPAWAR SSC-SD*
 - *useful demonstration software may be written*
 - *Core coming in FY99*
 - **Test suite under development**
 - *incremental releases began April 99; will continue*
 - *block functions, then views, then functionality, arithmetic, ...*

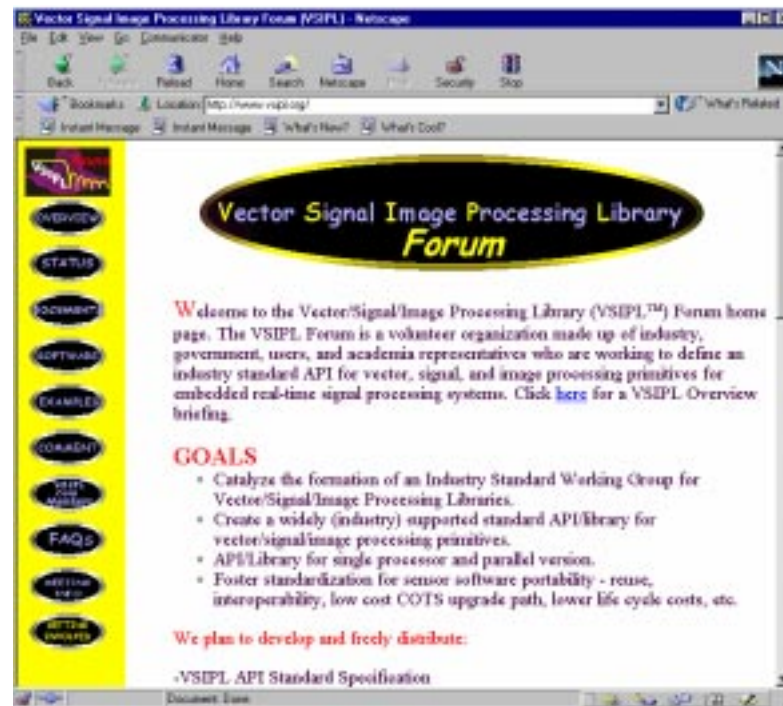


Interacting with the VSIPL Forum

- **E-mail reflector at <vsipl@list.gatech.edu>**
 - To subscribe: send to <listproc@list.gatech.edu> with following in body text:
 - *subscribe vsipl <your e-mail address> <your name>*
 - *e.g., subscribe vsipl mark.richards@gtri.gatech.edu Mark Richards*



- **VSIPL web site at <www.vsipl.org>**





Relation to TASP COE

- **The VSIP L API has been coordinated with the Navy Tactical Advanced Signal Processor (TASP), Common Operating Environment (COE) program.**
- **TASP COE continues to support VSIP L development**
 - maintain web site
 - test suite funding
 - integration testing and benchmarking



TASP DSP Common Operating Environment (COE)

- **Definition**
 - **Collection of Open Standards to Ensure Portability of Applications Software Among Various Implementations of the Standard Architecture**
- **Development Guidelines**
 - **All COE Standards Are Intended to be Commercial Standards**
 - *Must be Practical For Today*
 - *Requires Both Vendor and User Support*
 - *Need Vendor Implementations in TASP COE Time Frame*
 - **All COE Elements Should be Independent of Programming Language**
 - **Communications Standard Should be Independent of Any Vendor's RTOS**
 - **Government's Role is Facilitator**

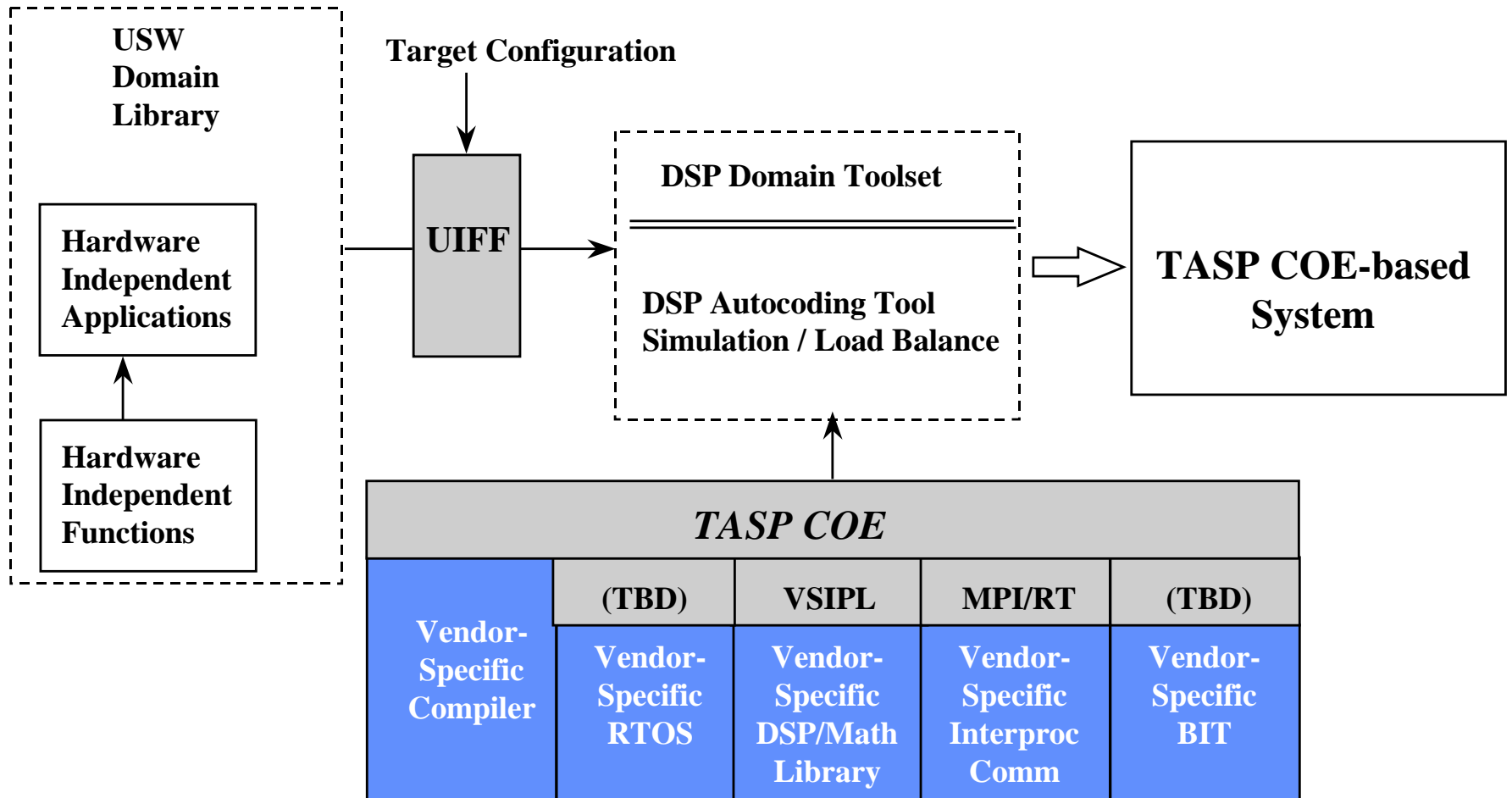


DSP COE Components

- **Near Term Components**
 - Language: ANSI C
 - **Standard Math/Signal Processing Library: VSIPL**
 - Communications: MPI/RT
 - RTOS Interfaces: TBD. (POSIX ??)
 - Performance Benchmarks: TBD
- **Longer Term Components**
 - Parallel Math Libraries
 - Built-In Test (BIT) Interface
 - Security
 - Fault Tolerance/Built-In Test Performance
 - Programming/System Development Tool Interfaces



TASP COE Software Library





TASP COE in FY99: Continuing VSIP L Development

- **For FY 99 TASP plans to**
 - **Finish VSIP L 1.0**
 - **Finish a VSIP L test suite**
 - **Finish a working demonstration model of MPI/RT with real time features on a multi-computer or network of workstations**
 - **Finish the Core version of the library useful for demonstration purposes**
 - **Produce some documentation for VSIP L Core to assist people who are investigating VSIP L**
 - **Look at integration issues of VSIP L and message passing on a network of workstations**
 - *Produce some VSIP L, MPI, and MPI/RT integration documentation*



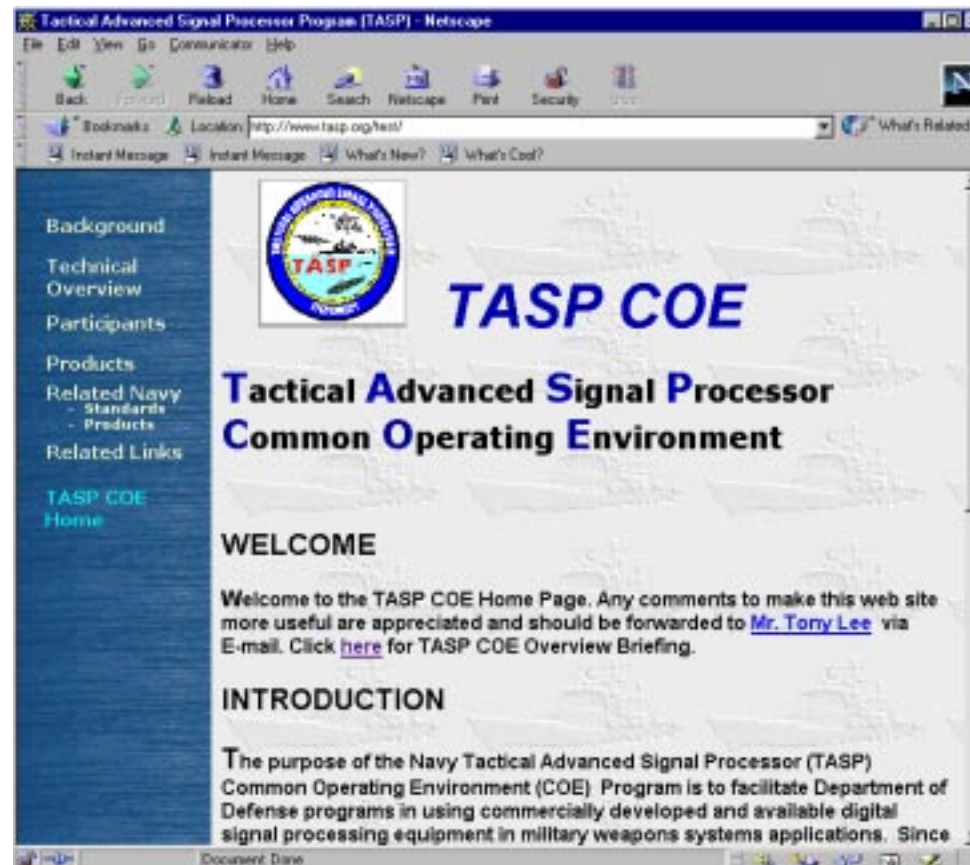
Current or Near-Term TASP COE & VSIPPL Products

- **TASP implementation of VSIPPL Core Lite library**
 - beta software developed by USN SPAWAR Systems Center, San Diego
 - extension to VSIPPL Core in progress
- **VSIPPL test suite**
 - incremental releases began April 1999
- **MPI/RT reference implementation due Fall 1999**



TASP COE Web Site

TASP COE Program can be tracked at <www.tasp.org>





What's Next for VSIPL Forum?

- **Standardization Activity**
 - After mature products appear on the market, formal standardization *may be* sought through IEEE (or ANSI)
 - Similar to the approach of the MPI Forum
- **Technical Activity**
 - C++ binding?
 - Parallel VSIPL?
 - VSIPL performance suite/benchmarking
 - *documenting the “price of portability”*
- **Need to Develop VSIPL & TASP COE Roadmap and Establish Support**



Next VSIP L Forum Meeting

- **July 13 & 14, 1999**
- **Mercury Computer Systems, Boston area**
- **Watch <www.vsipl.org> for details**