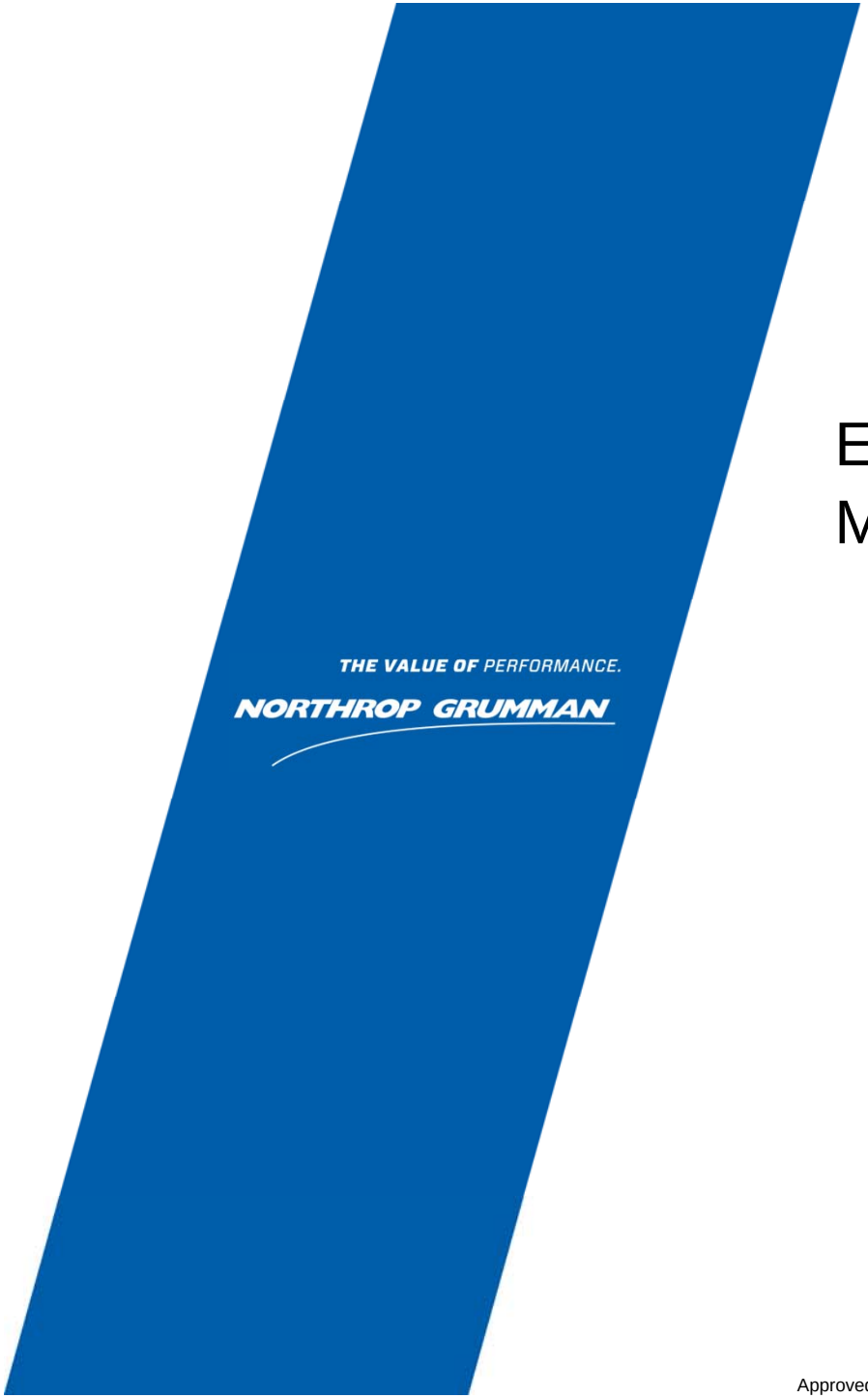


Expanding Legacy Systems Using Model Driven Engineering (MDE)

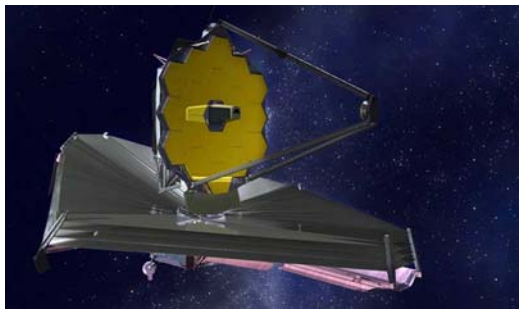
Kevin Nguyen & Billy Smith
Software Engineers

Overview

NORTHROP GRUMMAN

- Who we are
- MDE Process
- Results
- Challenges of expanding a legacy software system using MDE
- Lessons Learned

Northrop Grumman Products







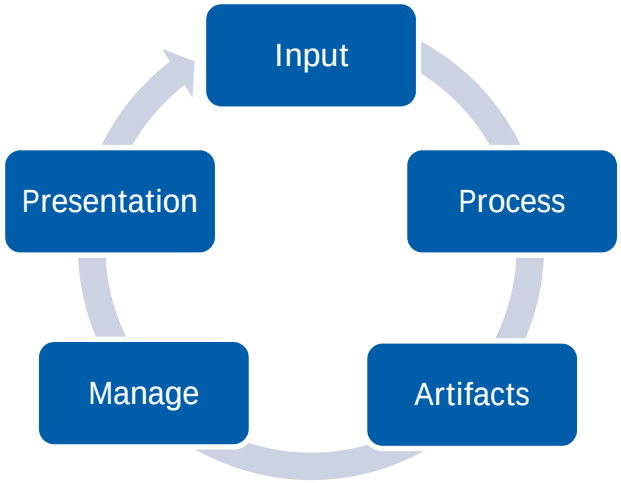



Model Driven Architecture Development Highlights

NORTHROP GRUMMAN

Design Phase	Purpose	Collaboration
System architecture	Understand customer's requirements	- System engineers work with customers to understand their requirements - Start a concept model to support the effort of breaking the system requirements into hardware, and software requirements
CSCI architecture	Create architecture of hardware and software	- System engineers work with senior architects and test engineers to convey knowledge as well as clarify any misunderstandings - Refine/derive requirements - Develop test plan
CSC architecture	Expand architecture to software components	- Senior architects work with subject matter experts (SME) to further break the architecture down into CSCs - Generate ICDs and data information - Refine test plan
Detailed Design	Develop software units	- SMEs work with software engineers to design and develop software units

Design Phase Lifecycle

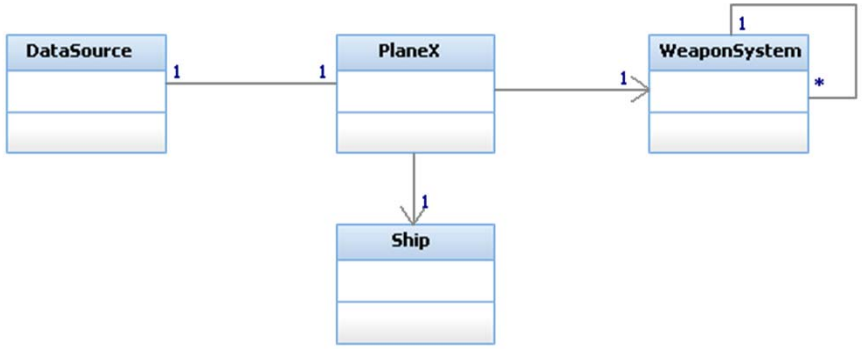


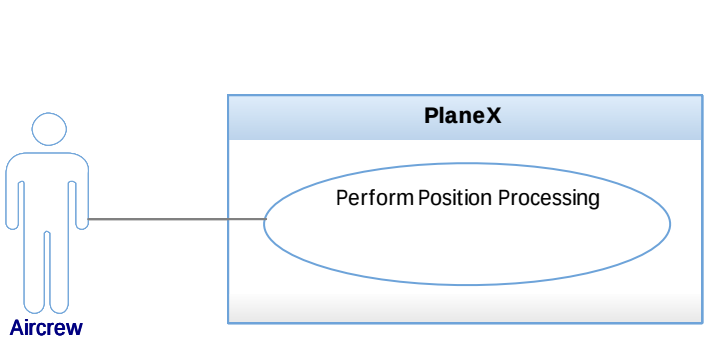
- Each phase of architecture development follows the same basic procedures
- Output from one phase becomes input for the next phase
  - Iterative process helps to catch design deficiency early

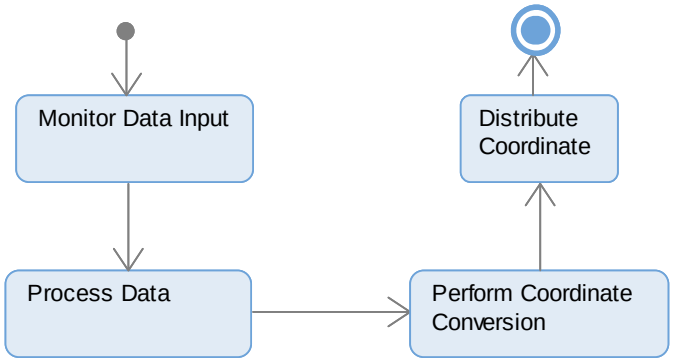
- Customers get the opportunity to participate in design and development efforts
- Architecture artifacts are developed using UML, making them easy for all parties to understand

Requirements Elicitation

- Inputs are from customers, users, and management

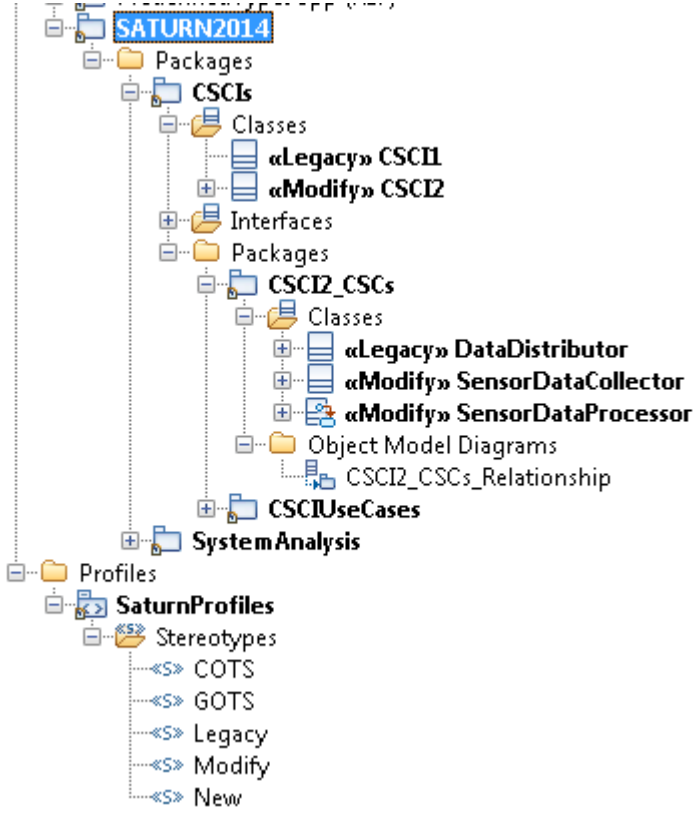


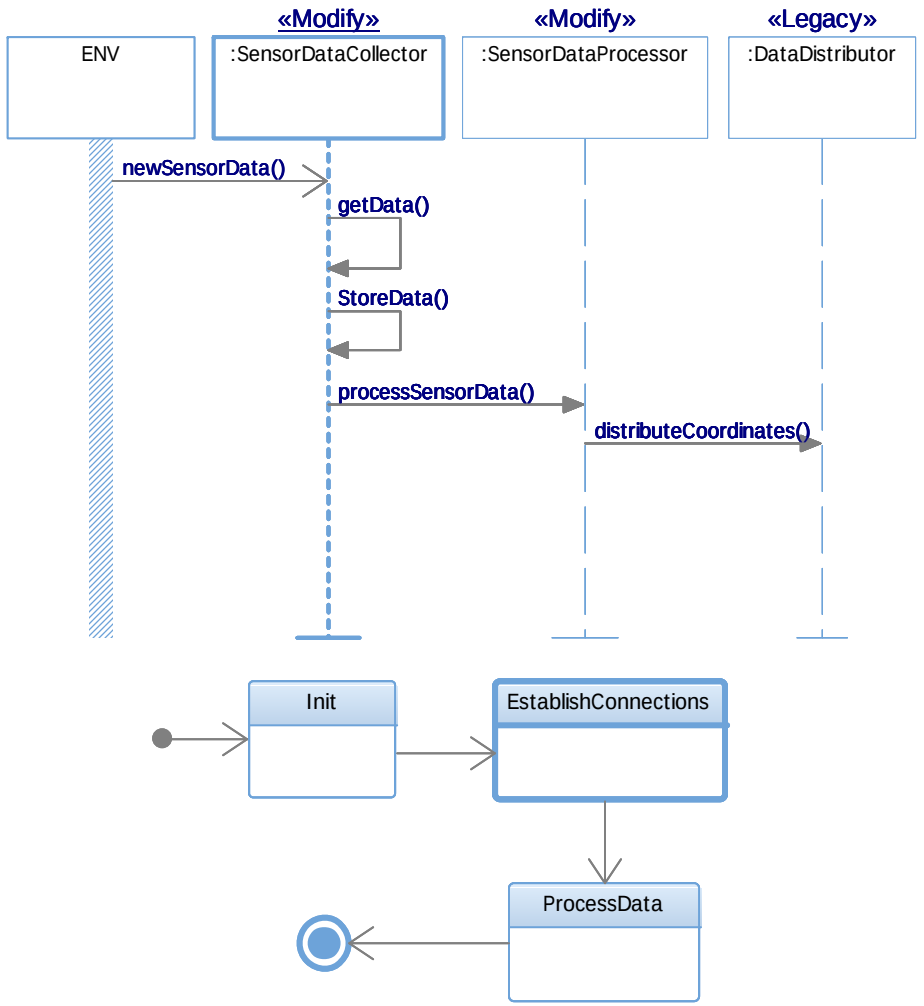




Understand Your Stakeholders

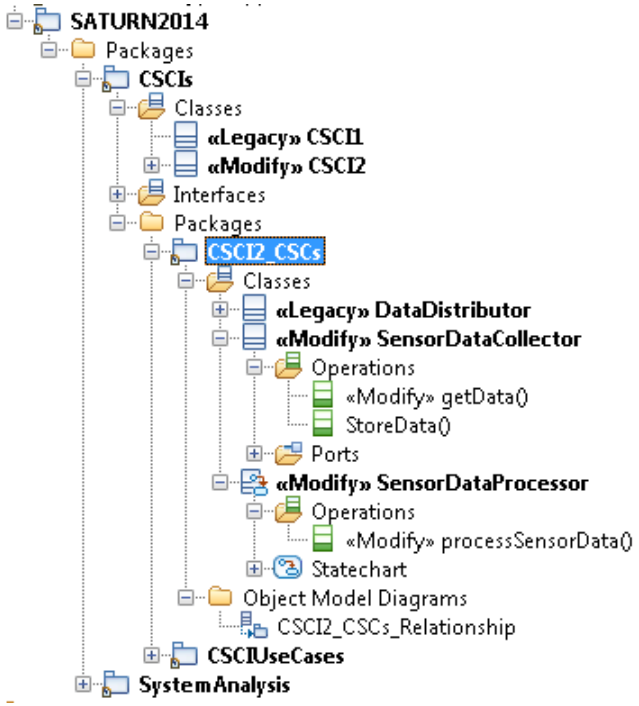
Requirements Allocation and CSCI Interfaces

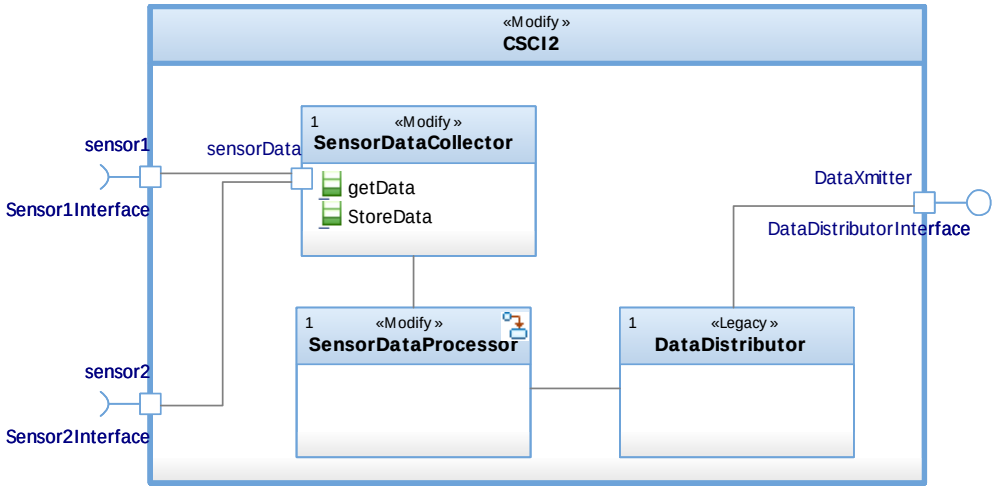


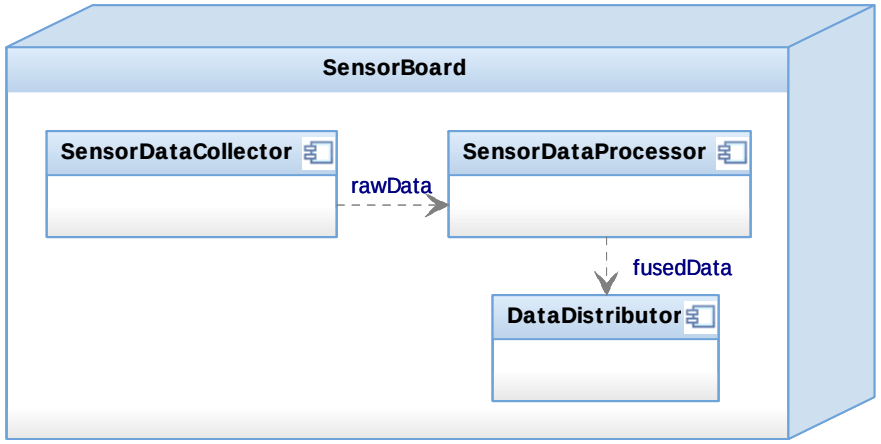


Decompose System Requirements to CSCI Level

Computer Software Component (CSC) Interfaces



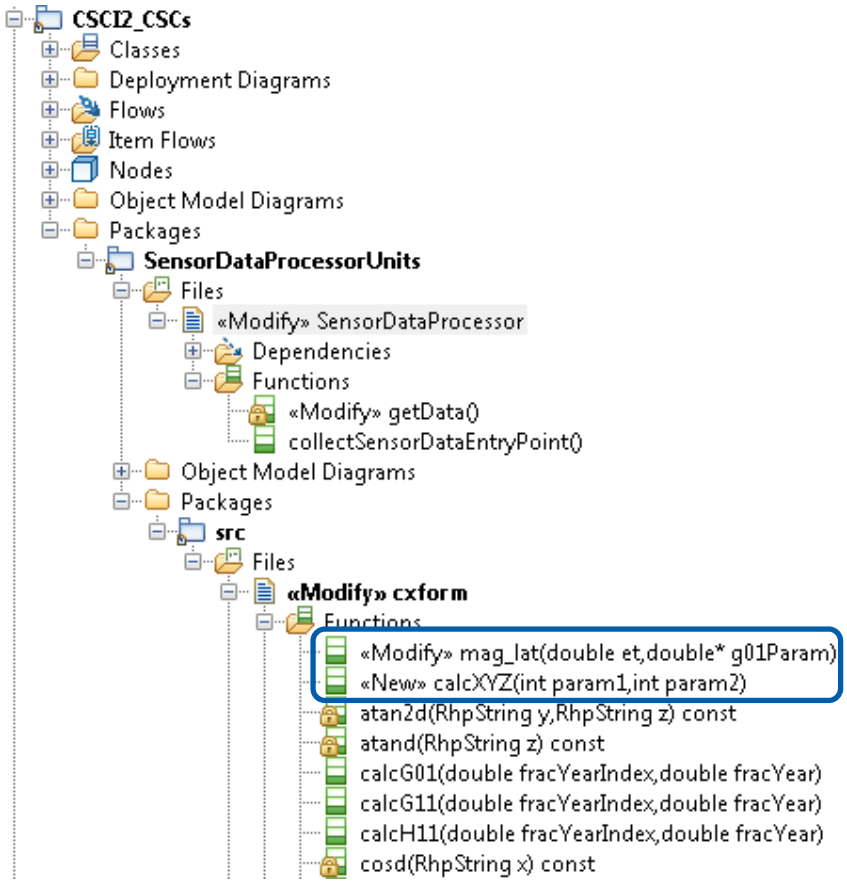


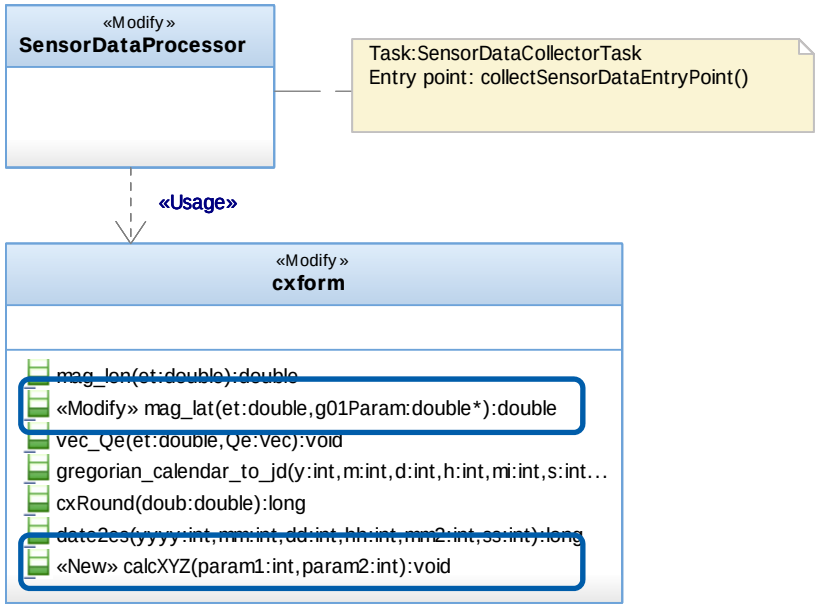


Decompose CSCI Requirements to CSC Level

Computer Software Unit (CSU) Detailed Design

NORTHROP GRUMMAN

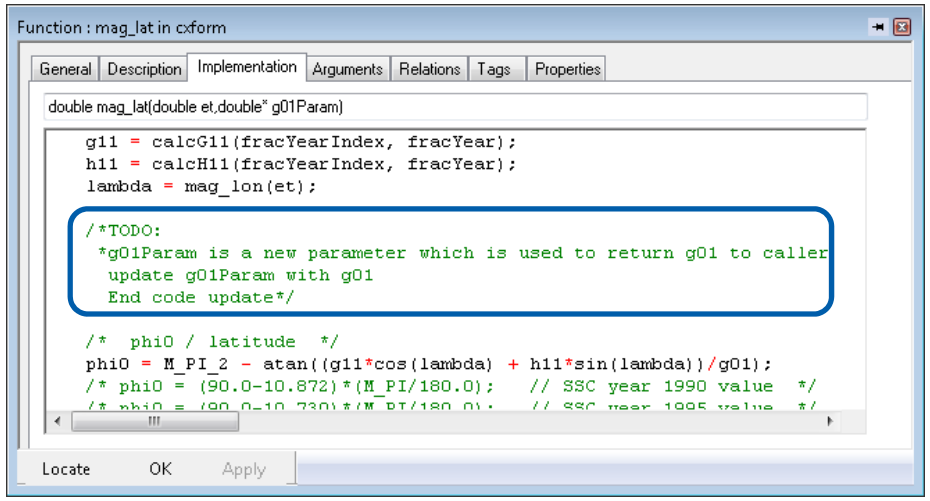


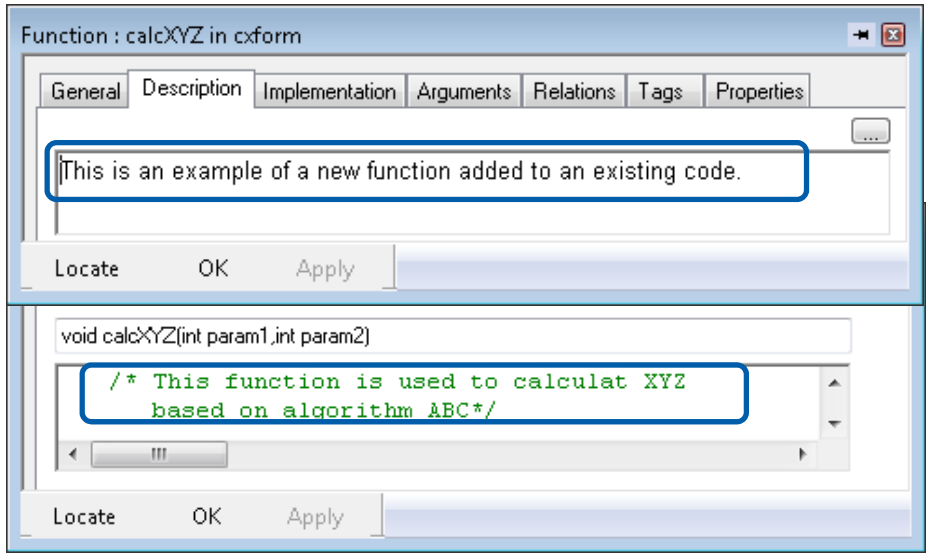


CXFORM package is downloaded from http://nssdcftp.gsfc.nasa.gov/selected_software/coordinate_transform/

Drive Changes to Software Units

Computer Software Unit (CSU) Detailed Design (continued)






```
double mag_lat(double et, double* g01Param) {
    double fracYearIndex = (et+3155803200.0)/157788000.0;
    double fracYear = fmod(fracYearIndex, 1.0);
    double phi0, lambda, g01, g11, h11;

    if (fracYearIndex >= NUM_IGRF_YEARS_DEFINED)
    {
      /*  fprintf(stderr, "ERROR: Specified year is greater than IGRF im
      exit(EXIT_FAILURE);
    }

    g01 = calcG01(fracYearIndex, fracYear);
    g11 = calcG11(fracYearIndex, fracYear);
    h11 = calcH11(fracYearIndex, fracYear);
    lambda = mag_lon(et);

    /*TODO:
     *g01Param is a new parameter which is used to return g01 to caller
      update g01Param with g01
      End code update*/

    /*  phi0 / latitude  */
    phi0 = M_PI_2 - atan((g11*cos(lambda) + h11*sin(lambda))/g01);
    /* phi0 = (90.0-10.872)*(M_PI/180.0);   // SSC year 1990 value  */
```

cxform.c | cxform.h

```
void mat_transpose(Mat m_in, Mat m_out);
void mat_times_mat(Mat m1,   Mat m2, Mat m_out);
void mat_times_vec(Mat m1,   Vec v1, Vec v_out);

enum direction { FORWARD, BACK };
typedef enum direction Direction;
/* This is an example of a new function added to an existing code. */
void calcXYZ(int param1, int param2);

#endif
```

cxform.c | cxform.h

```
#define M_PI_2 1.57079632679489661923
void calcXYZ(int param1, int param2) {
    /* This function is used to calculat XYZ
       based on algorithm ABC*/
}
```

CXFORM package is downloaded from http://nssdcftp.gsfc.nasa.gov/selected_software/coordinate_transform/

Results

NORTHROP GRUMMAN

- Passed PDRs and CDRs with full award fee and minimal customer Request for Information (RFI)
- Created a framework for collaboration and open communication between system, software, test, architects, and customers
- Minimized software integration issues
- Increased software testability
- Obtained the benefits of object oriented design and analysis for new architecture built on top of a non-OO system
  - Reusability, testability, maintainability
  - Highly cohesive, low coupling

Challenges of Adapting MDE to Legacy System

NORTHROP GRUMMAN

- Acquiring new tools and training
  - UML tools, process training
- Integrating the MDE process into existing tools and processes
  - Source control, requirements management, in-house tools
- Gaining full commitment and support from leadership
  - Selling the long term benefits of using MDE
- Changing the existing culture
  - Removing the engineering group stovepipe
  - Partitioning legacy software components
  - Stubborn engineers

Lessons Learned

NORTHROP GRUMMAN

- Allocate enough time for design
  - Spend more time on algorithm analysis
  - Perform more trade analysis
- Prototype often
- Fully flesh out interfaces early on
- Revisit and evaluate use cases at every design level
  - Make sure critical design elements are not eclipsed or forgotten
- Do not underestimate the effort of changing the existing culture


THE VALUE OF PERFORMANCE.
NORTHROP GRUMMAN