Allegro CL®

Success Story

● ● ● ● ● ● ● ●

Johnson Engineering
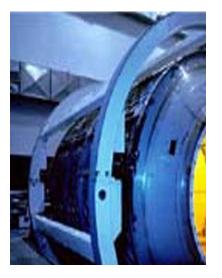### Packing Cargo for the new International space station is now easier…Thanks to Lisp and Allegro CL.

Johnson Engineering packs the Multi-Purpose Logistics Modules (MPLM) used to hold the cargo that's shipped to the International Space Station via the Space Shuttle. The modules are massive and can hold up to 20,000 pounds of cargo. Configuring the contents of, and actually packing these modules is a challenge.

Packing the MPLM was previously done using a combination of CAD tools and spreadsheets. The process is now managed, however, with a Common Lisp application called the Cargo Planning and Analysis System (CPACs).

The CPACS system represents a significant time and financial savings for the project. Prior to CPACs, it took four people two weeks to do initial configuration and planning — and that didn't take into consideration any weight balance issues or cargo modifications.

Using the CPACs application however, one person can come up with an initial configuration in about six hours.
parts: user profiling, content profiling and matching.

Changes can now be easily incorporated into the manifest — CPACS simply recalculates everything automatically and identifies any balance problems that might arise from the new configuration.

● ● ● ●

"Lisp's code-compile-debug cycle means that I can rewrite a function and instantly see how it affects the application."

Jason Kantz,
Johnson Engineering

The CPACs application was originally created at the Boeing Company for NASA. It was used in a minor capacity for several years until an engineer at Johnson Engineering brought it to the attention of the Crew Station Operations Division.

This division reviewed the application and determined that, when reconfigured to support current standards and specifications, it would have the potential to eliminate much of the workload associated with stowage integration. Thanks to the capabilities inherent in Lisp— like automatic memory management, manifest typing, first class functions, and closures—it would be significantly easier to bring CPACs up to date than if it had been built in another language.

Jason Kantz, an employee at Johnson Engineering, was sent to Franz Inc. with the assignment of learning common lisp and rebuilding the application to support current requirements. He would need to add NASA's new stowage provisions and integrate them with the auto-packing procedures and mass-properties calculations in CPACs. Plus, CPACs would also need to import manifest characteristics data from the Vehicle Master Database for thousands of parts.

Although he had programmed in Scheme before, Jason had never used Lisp. He knew that he needed to become proficient in the language before CPACs could be successful. He especially wanted to make the application easier to debug, and also wanted to better understand how it was designed.

"During the month of training, I learned to take advantage of every timesaving feature of the language and development environment." Says Kantz. "…By invoking the debugger, I learned to inspect the call stack of the running application, automatically jump to the source code of the problem procedure,

incrementally compile the correction, and finish without ever having to restart or recompile any other part of the application. "

Jason focused his learning efforts on the Lisp and Allegro CL tools and features that would help him understand how CPACs worked — such as the Franz Composer Software, which was used to graph class hierarchies.

By learning and using Allegro CL, he was able to create specialized methods that organized the behavior of objects within the application.

"Organizing the procedures for these special cases would not have been so simple in a language that uses static typing and a message-passing model for object methods. Without generic functions, I would have spent less time writing useful code and more time trying to figure out how to perform this dispatch on multiple types." Adds Jason

Kantz now prefers Lisp over any other computer language and believes using it makes applications much more flexible and easy to design especially for someone getting up to speed in a new language. "Lisp's code-compile-debug cycle means that I can rewrite a function and instantly see how it effects the application."

Kantz continues to make updates and changes to the application, which will now be used for all future space station re-supply missions. He's currently rewriting the application to remove many of the pre-specified size and layout constraints. Once again, CLOS makes this easy to do because it enables him to design a generic vs. specific model.

For more information on the CPACs application, visit http://kantz.com/jason/cpacs.html