

Designed for Multiprocessors

Gedae simplifies the complexities of programming multi-core signal and data processing applications by providing a development environment for multiprocessor and multicore systems.

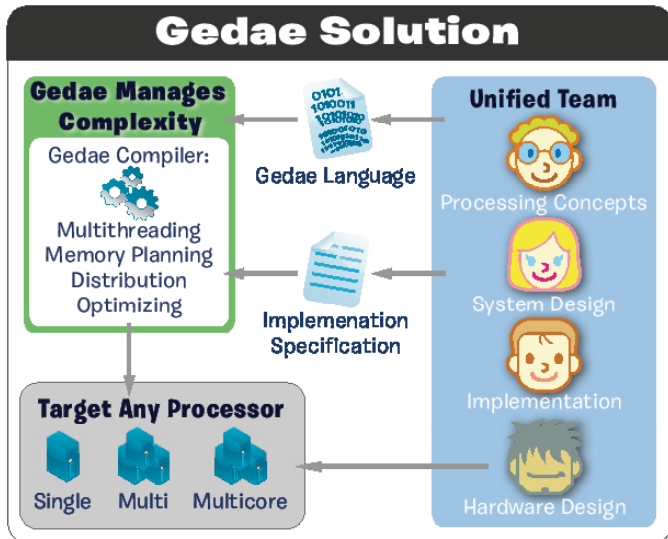
Gedae automates the most complex and tedious parts of the development process to maximize efficiency. Developers can specify the functionality of an application and map it to a hardware architecture in a fraction of the time previously required. Gedae accommodates user control of automation. Developers observe the behavior of an application and adjust the processing mapping, interprocessor communications, granularity, buffering, memory mapping, thread management and executable structure.

Gedae also helps optimize development by building hand-coded efficient applications. It provides tools to ease development, improve productivity, increase the pool of qualified developers, and reduce the development cycle and lifetime maintenance costs.

What would a true multiprocessor solution look like?

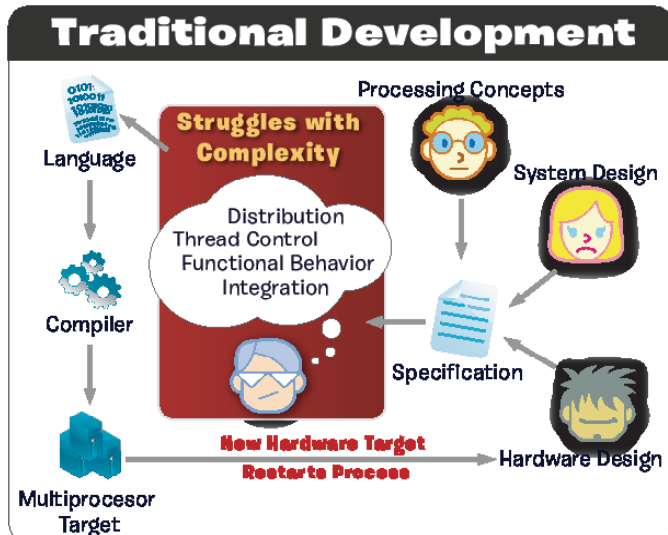
- Language syntax and semantics = functional specs
- Integrated language and editor
- visible implementation
- true multiprocessor compiler
- Can build any application
- Targets complete system
- Execution analysis tool
- distributed debugging tools

GEDAE is the only solution!

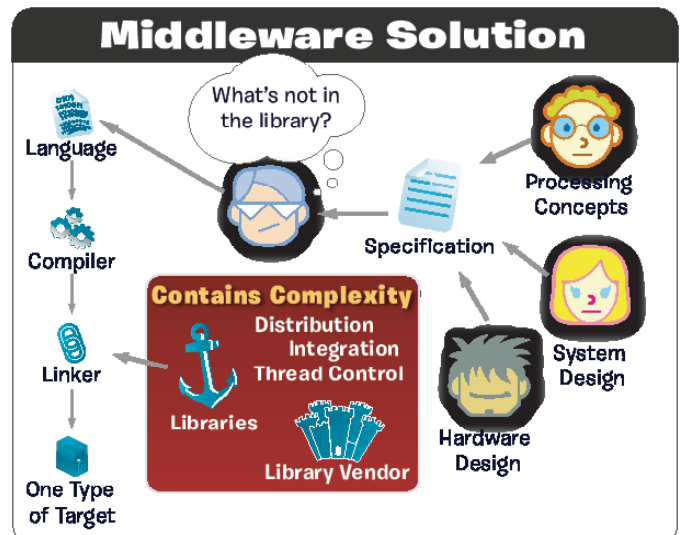


How is Gedae different? The Gedae compiler handles the complexity. Since the Gedae compiler is aware of the multi-processor and the multi-core and memory architecture of the target, it automatically customizes the software to that architecture. As a result, it's easy to port applications among processors.

The user supplies the implementation parameters and the Gedae language provides the compiler with information in a form that allows the compiler to automatically perform the most complex tasks, including threading the application, planning memory and implementing the distribution of the software.



In a traditional development process systems engineering provides the specification of the system and software and the development team is responsible for all the complexity of the system.



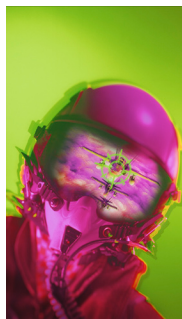
Middleware attempts to simplify development by providing a HW independent method of specifying common operations shifting the complexity problem to the library. A key problem of reuse libraries is a difficulty of scaling in both component size, feature variations and platforms targeted.

Key Features and Benefits

Feature	Benefit
Gedae Language	
Block Diagram Programming Language	Component based language promotes software reuse and provides basis for partitioning and mapping to multiple processors
Hierarchy, Family and Route Boxes	Data distribution and parallelization is easily parameterized and expressed in the block diagram
Data Flow Graph Language	An intuitive way of expressing software control that provides the basis for many compiler optimizations such as threading and mode control
Gedae Compiler	
Threading	Automatically group components into threads, where the order of execution for the components in each thread are preplanned
Partitioning and Mapping	Easily distribute and redistribute processing across multiple processors, automatically incorporating the necessary communication into the threads
Launch Package	One step creation of stand alone set of executables, completely divorced from the development environment, along with options for launching from EE-PROM, executable compression, and other features
Analysis	
Analysis Tools	Execution analysis (including timing, timeline, order, and memory usage), data flow and distributed debugging, parameter trace

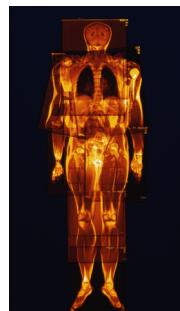
Industries

Defense



The modern battlefield is controlled as much by compute power as firepower. The weapons and vehicles of modern warfare require multiple computer systems and significant computing power in order to operate and react in real time. Audio, image, sonar, radar, and other signals are processed by algorithms that perform target recognition, tracking, encryption, speech recognition, and many more applications. Gedae can be used to harness the compute power necessary to implement these start-of-the-art defense applications while shortening time to market and combating obsolescence.

Medical Imaging



The use of computing is now an essential part of both medical research and diagnostic medicine, using software to enhance and classify images from MRIs, CT scans, ultrasounds, and radioactive sources. Image processing functions such as filtering, feature extraction, image segmentation, and visualization provide a powerful method for viewing affected areas of the body which were difficult or even impossible to view in previous years. Gedae enables medical imaging software to more effectively utilize high performance computers, allowing physicians and biologists to process and manipulate these images in real time, shortening the gap between test and diagnosis.

You imagine it, Gedae will help you build it!



Gedae is applicable to a wide range of applications, from commercial audio processing toolboxes for personal computers to onboard software driving the microcontrollers of an automobile. Essentially any application that processes a stream of data can be developed in Gedae to take advantage of its benefits in productivity, efficiency, portability and obsolescence proofing.

General Specifications

Supported Architectures:

Single, Multi or Multicore Processors

Supported Vendors:

IBM, Mercury, Curtiss-Wright, Bittware, Custom

OS: Windows, Linux or Solaris

Requirements: C Compiler, X Server, Motif Libraries

The Windows version uses Exceed with XDK to provide the X server and Motif libraries



1247 N Church Street, STE 5, Moorestown, NJ 08057 USA P:(856) 231-4458 F:(856) 231-1403

www.gedae.com

e-mail: gedae@gedae.com