

Industrial attempts to create a software platform for parallel programming is in the initial stages: first products were distributed in late 2007, mostly for military clients, more details are presented in the website www.gedae.com. GEDAE is an US company and even if it will be successful in “parallelization” of new software applications it is hardly unlikely that US products intensively used for military applications will be widely available for European companies.

The following table compares GEDAE and ERA proposed approaches.

Software Industrial vision vs. ERA proposal

Industrial state of the art (GEDAE)	Proposed by ERA	Comment
Block diagram programming language.	New, compact and efficient language with no deliberate parallel descriptions, if possible.	ERA proposes <i>automatic</i> analysis of algorithm parallelism and via <C,D,P> graphs and GLM form maximum possible parallelism.
Hierarchical tree organization of the programs.	Automatic structuring of the program to maximize parallelism.	GEDAE supposes user involvement on program tree formation.
Data distribution and parallelization is introduced manually.	Automatic and flexible memory and processors resource allocation using ERA HW and SW design.	Minimum concurrency will be handled dynamically.
Specially developed memory partitioning.	Flexible partitioning, including dynamic reconfiguration of the resources for performance, reliability (fault tolerance) and power consumption.	ERA hardware design will be patented.
Organization of sequences for program dependencies and task sequences.	ERA graph-logic model is unique and provide maximum efficiency HW and SW resources use at run-time level.	ERA rigorous design and application of GLM both introduce new paradigm for computation.
Prescheduled program execution.	Re-configurability of task scheduling and recoverability of procedure/module. Task at the level of run time, prepared through compiling.	
Support of standard concurrency via parameters triggering and state machine mechanism.	New fault tolerant synchronization schemes; GLM modelling and design of software with minimum concurrency and maximum parallelism.	Patentable method.
Support: parameter tracing, control (in terms of time and sequence) program tracing.	Graph theory schemes with consistency of all three program graphs: control, data and predicate dependencies.	Main graph theory results might be a theoretical core of new debugging tool for ERA.