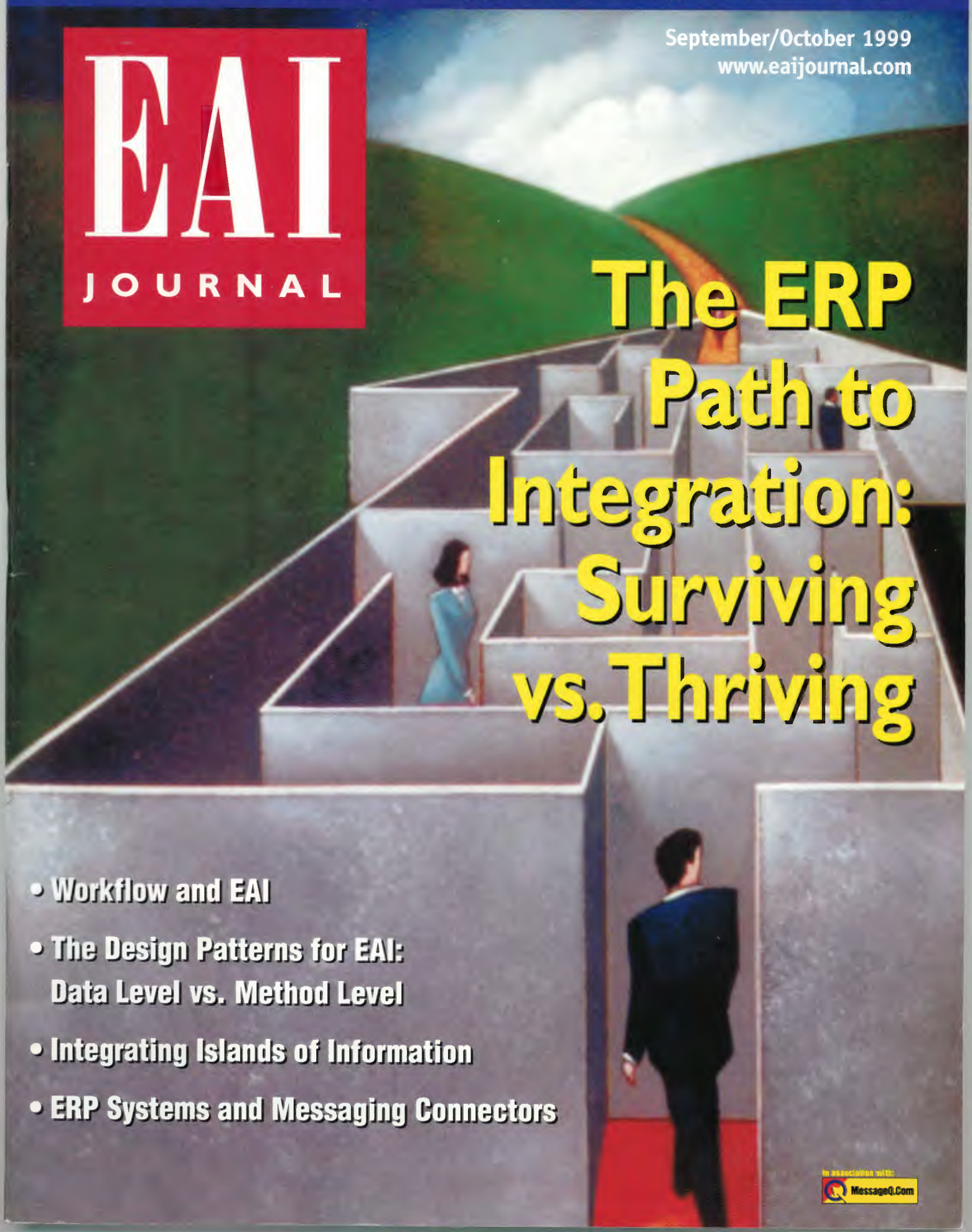
The Source for Enterprise Application Integration Solutions

EAI JOURNAL

September/October 1999
www.eaijournal.com



The ERP Path to Integration: Surviving vs. Thriving

- Workflow and EAI
- The Design Patterns for EAI: Data Level vs. Method Level
- Integrating Islands of Information
- ERP Systems and Messaging Connectors

enterprise integrity

By DAVID McGOVERAN

Think BPMS!

Remember the days when data was managed exclusively by applications? Each application owned the formats, operations, and I/O methods. Oh, the long nights we used to spend examining COBOL or FORTRAN (yes, all caps — they're acronyms) code for data semantics just to write a meaningful new report program. Wonder why we didn't just look the information up in the data dictionary, repository, or system catalog; or why the programmers didn't just use a DBMS? For all of you that take DBMS concepts and facilities for granted, the answer is — they didn't exist! DBMS technology provides *consistent, managed data change*, the essence of business transaction processing. Its value to EAI is well-known, even if seldom acknowledged.

Lessons can be learned from the DBMS story and a new opportunity seized in doing so. RDBMSes brought a true IT revolution, changing computing forever. When commercial RDBMSes were introduced, there were no standard ways of (1) organizing data, (2) accessing data interactively, (3) accessing data programmatically (despite CODASYL, proprietary approaches ruled!), or (4) remotely accessing data (files yes, data no). RDBMS availability made data sharing (across modules, applications, and even systems) possible and ultimately the norm.

Of course, it took a while to get a standard language and a non-proprietary API even longer. The relational model made both possible: physical data independence hides the dirty details of physical storage, logical data independence hides logical changes where desired, and location independence hides physical location. Without this encapsulation, distributed applications were impractical: it is no accident that client/server and network computing's predecessor (LAN-based computing) took off with RDBMSes. And without these, EAI (and ERP) could not be.

The great new opportunity I mentioned? Well, today applications hold processes captive much as they once did data. There is no standard process language or API, and no process analogue of the RDBMS. So last year I suggested a new concept to several EAI vendors — the BPMS (Business Process Management System). A BPMS manages business processes *independent of applications*: process information is easily shared, accessed, and managed. As data modeling tools are used to design database schemas, so process modeling tools could be used to design process schemas (improperly called "processes"). Like an RDBMS, a BPMS stores process schemas, process constraints, and inter-process relationships in its system catalog. A repository for process "runs" the BPMS analogue of RDBMS data.

Early commercial RDBMSes provided tremendous and immediate investment return by enhancing query, decision support, and report capabilities, as will the BPMS. And the decision support/analytic potential of a BPMS can take us far beyond querying the "status" of some business process, allowing physical performance parameters of a process schema to be related to *business performance metrics*.

Even so, such passive BPMS uses ignore the real potential as active IT components. BPMSes will be more than process schema repositories, becoming the process "database of record," and externalizing business function logic. The BPMS then both monitors *and* controls process flow much like an intelligent message router: after all, a business process is just a set of interconnected business rules, a network of business events, activities, or functions. At the highest level of process abstraction, the BPMS controls the business logic between applications, business functions, or activities, permitting process modification without shutting applications down.

Today's workflow products can be considered a primitive BPMS technology, but lack key features. General process schemas, real-time performance, and transactional recovery features must be supported to obtain the real potential of BPMSes. That potential lies in process independence analogues of physical, logical, and location independence which are necessary for EAI success. Like data schemas, process schemas (found today as application control or procedural logic) must support multiple descriptive levels or *process abstraction hierarchies*. Unfortunately, the procedural logic that implements multiple process levels is rarely separated in application code. Solve these problems, and BPMSes will enable changes to the physical implementation of business logic without impacting management's business process understanding. More important, managers can then implement business process changes with minimal impact on, or involvement from IT.

Early BPMS technology exists in products like Hewlett Packard Changengine, IBM MQWorkflow, Vitria, and others. Still, analogues of SQL, database transactions, and relational API are needed to progress. And we need dynamic optimization of the location of process execution — whether externalized in a BPMS process engine or internally in the application. Although BPMS technology lacks the functional maturity or definition associated with RDBMSes, today's products tangibly hint at the flexibility and economic benefits of the BPMS vision. EAI's IT and business drivers foretell increasingly rapid business process change. Without a BPMS as a key component of the EAI architecture, that demand will not be met.

And I hope you'll agree that *consistent, managed business process change* is the very essence of enterprise integrity — and therefore of EAI.

David McGoveran is president of Alternative Technologies, Inc. He has more than 20 years' experience with mission-critical applications and has authored numerous technical articles on application integration. E-mail: mcgoveran@alternativetech.com.

Discuss "Enterprise Integrity" with David McGoveran at www.eaiforum.com.