Architectural pattern

An **architectural pattern** is a general, reusable solution to a commonly occurring problem in software architecture within a given context.[1] Architectural patterns are similar to software design patterns but have a broader scope. The architectural patterns address various issues in software engineering, such as computer hardware performance limitations, high availability and minimization of a business risk. Some architectural patterns have been implemented within software frameworks.

Contents

Definition

Even though an architectural pattern conveys an image of a system, it is not an architecture. An architectural pattern is a concept that solves and delineates some essential cohesive elements of a software architecture. Countless different architectures may implement the same pattern and share the related characteristics. Patterns are often defined as "strictly described and commonly available".[2][3]

Architectural style

Following traditional building architecture, a 'software architectural style' is a specific method of construction, characterized by the features that make it notable".

> *An architectural style defines: a family of systems in terms of a pattern of structural organization; a vocabulary of components and connectors, with constraints on how they can be combined.*[4]

> *An architectural style is a named collection of architectural design decisions that (1) are applicable in a given development context, (2) constrain architectural design decisions that are specific to a particular system within that context, and (3) elicit beneficial qualities in each resulting system.*[1]

Some treat architectural patterns and architectural styles as the same,[5] some treat styles as specializations of patterns. What they have in common is both patterns and styles are idioms for architects to use, they "provide a common language"[5] or "vocabulary"[4] with which to describe classes of systems.

The main difference is that a pattern can be seen as a solution to a problem, while a style is more general and does not require a problem to solve for its appearance.

Examples

Here is a list of architecture patterns, and corresponding software design patterns and solution patterns.

Sub-domain area	Architecture pattern	Software design patterns	Solution patterns	Related patterns
Data integration/SOA	▪ ETL (data extraction transformation and loading)	▪ Change data capture ▪ Near real-time ETL ▪ Batch ETL ▪ Data discovery	▪ Error handling ▪ Job scheduling ▪ Data validation ▪ Slowly changing dimensions load	▪ EAI ▪ Master data hub ▪ Operational data store (ODS) ▪ Data mart ▪ Data warehouse
	▪ MFT			
	▪ EAI/ESB	▪ Publish/subscribe ▪ Request/reply ▪ Message exchange patterns	▪ One-way ▪ Synchronous request/response ▪ Basic callback ▪ Claim check	▪ SOA
Data architecture	▪ Transaction data stores (TDS/OLTP) ▪ Master data store ▪ Operational data store ▪ Data mart ▪ Data warehouse	▪ Custom applications databases ▪ Packaged application databases		▪ ETL ▪ EAI ▪ SOA
Analytics and business intelligence	▪ Transactional reporting ▪ Operational analytics ▪ Business analytics ▪ Predictive analytics ▪ Prescriptive analytics ▪ Streaming analytics ▪ Data science and advanced analytics ▪ NLP	▪ Transactional reporting data access ▪ Operational reporting data access ▪ Analytical reporting data access ▪ Analytical dashboard data access ▪ Operational dashboard data access ▪ Data mining	▪ Real-time dashboards ▪ In-memory analytics ▪ Statistical analysis ▪ Predictive analytics	▪ ETL ▪ EAI ▪ TDS ▪ Operational data store ▪ Data mart
Master data management	▪ Master data hub	▪ Master data replication ▪ Master data services ▪ Master data synchronization		▪ Change data capture ▪ EAI ▪ STD
Data modeling	▪ Dimensional data modeling ▪ E-R data modeling	▪ Modeling standards ▪ Naming conventions		
Artificial intelligence	▪ Decision management	▪		

	▪ Speech recognition ▪ Text analytics and NLP ▪ Natural language generation ▪ Classic machine learning ▪ Deep learning ▪ Robotic process automation ▪ Image and video analysis	▪		

Some additional examples of architectural patterns:

- Blackboard system
- Broker pattern
- Event-driven architecture
- Implicit invocation
- Layers
- Microservices
- Model–view–controller, Presentation-abstraction-control, Model-view-presenter, and Model-view-viewmodel
- Entity–component–system
- Multitier architecture (often three-tier or n-tier)
- Naked objects
- Operational data store (ODS)
- Peer-to-peer
- Pipe and filter architecture
- Service-oriented architecture
- Space-based architecture

See also

- List of software architecture styles and patterns
- Process Driven Messaging Service
- Enterprise architecture
- Common layers in an information system logical architecture

References

1. R. N. Taylor, N. Medvidović and E. M. Dashofy, Software architecture: Foundations, Theory and Practice. Wiley, 2009.
2. Chang, Chih-Hung; Lu, Chih-Wei; Lin, Chih-Hao; Yang, Ming-Feng; Tsai, Ching-Fu (June 2008). "An Experience of Applying Pattern-based Software Framework to Improve the Quality of Software Development: 4. The Design and Implementation of OS2F" (https://web.archive.org/web/20110922035257/http://jses.seat.org.tw/index.php/jses/article/viewFile/41/30). *Journal of Software Engineering Studies, Vol. 2, No. 6*. the Third Taiwan Conference on Software Engineering (TCSE07). pp. 185–194. Archived from the original (http://jses.seat.org.tw/index.php/jses/article/viewFile/41/30) on 2011-09-22. Retrieved 2012-05-16. "Furthermore, patterns are often defined as something "strictly described and commonly available". For example, layered architecture is a call-and-return style, when it defines an overall style to interact."
3. "Architectural Patterns: Definition" (https://web.archive.org/web/20120623081009/http://aahninfotech.com/arct_pattern.html). AAHN INFOTECH (INDIA) PVT LTD. Archived from the original (http://aahninfotech.com/arct_pattern.html) on 2012-06-23. Retrieved 2012-05-16. "Even though an architectural pattern conveys an image of a system, it is not an architecture as such. An architectural pattern is rather a concept that solves and delineates some essential cohesive elements of a software architecture. Countless different architectures may implement the same pattern and

thereby share the related characteristics. Furthermore, patterns are often defined as something "strictly described and commonly available"".

4. M. Shaw and D. Garlan, Software architecture: perspectives on an emerging discipline. Prentice Hall, 1996.
5. http://msdn.microsoft.com/en-us/library/ee658117.aspx

Bibliography

- Avgeriou, Paris; Uwe Zdun (2005). "Architectural patterns revisited: a pattern language". *10th European Conference on Pattern Languages of Programs (EuroPlop 2005), Irsee, Germany, July*.
- Buschmann F.; Meunier R.; Rohnert H.; Sommerlad P.; Stal M. (1996). *Pattern-Oriented Software Architecture: A System of Patterns*. John Wiley & Sons.
- Bass L.; Clements P.; Kazman R. (2005). *Software Architecture in Practice: Second Edition*. Addison-Wesley.