Arquitectura de Sistemas de Visualización

Jorge Ardenghi Silvia Castro Elsa Estévez Pablo Fillottrani Sergio Martig Marisa Sánchez
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur
{jra,smc,ece,prf,srm,mas}@cs.uns.edu.ar
Tel. +291-4595135 Fax +291-4595136

Introducción

Mientras las nuevas tecnologías tanto en hardware como en software contribuyen a un aumento en el volumen y la diversidad de los datos que las organizaciones manejan, la exploración y visualización de estos datos se tornan cada vez más dificultosas. Por esto, y debido al crecimiento en la cantidad y variedad de usuarios y a la demanda de mayor funcionalidad, es deseable que los ***Sistemas de Visualización de Información*** (VI) cumplan con propiedades como la extensibilidad, la modificabilidad, la predictibilidad y la posibilidad de estar integrados por componentes intercambiables. Numerosos ambientes de VI han sido presentados en los últimos años; además, existen paquetes estándares que implementan la mayoría de las técnicas de visualización conocidas. A pesar de que estos sistemas no cumplen completamente con las propiedades mencionadas, sus características similares permiten afirmar que el área está madurando lo suficiente como para desarrollar una arquitectura de referencia que sí facilite su cumplimiento. Tal arquitectura de referencia permitiría la especificación e integración de los componentes de hardware y software necesarios para la implementación de sistemas de VI para los diferentes dominios que han surgido recientemente, tales como las consultas a bases de datos, data mining, monitoreo de sistemas, etc.

En el marco del Proyecto a desarrollar se propone definir una arquitectura de referencia que pueda facilitar la implementación de ambientes de visualización extensibles y modificables. Posteriormente se proyecta concretar estas ideas en un prototipo de ambiente de visualización, en base a algunos de los dominios de aplicación particulares. La tarea a desarrollar involucra una estrecha colaboración de los grupos de investigación en Computación Gráfica y Visualización, Ingeniería de Software y Sistemas Distribuidos del Departamento de Ciencias e Ingeniería de la Computación de la Universidad Nacional del Sur.

Contexto de la Investigación

La visualización y la exploración de colecciones de información se tornan cada vez más dificultosas a medida que el volumen y la diversidad de los datos aumenta [2]. Esto exige que, principalmente, los ***Sistemas de Visualización de Información*** (VI) deban ser extensibles, teniendo en cuenta además el crecimiento en la cantidad y diversidad de los usuarios y la creciente demanda para tener mayor funcionalidad. Por lo tanto, se requiere de ***ambientes de visualización*** que puedan adaptarse tanto a las necesidades de los usuarios como a las de los diseñadores de las técnicas de visualización para los distintos dominios de aplicación. Otras propiedades deseables son la reusabilidad de sus componentes, la facilidad de uso, la compatibilidad con otros ambientes y la portabilidad.

Actualmente hay varios paquetes de ***VI***, tales como Xmdv, Xgobi, IBM Parallel Visual Explorer, que se aplican a diversas fuentes de datos. Si bien éstos poseen específicamente capacidades de análisis y visualización, tienen el inconveniente de no ser flexibles y no tener una buena interacción con el mundo exterior. Por otro lado, los sistemas de Visualización como AVS, IBM Data Explorer y NAG Iris Explorer le dan al usuario un ambiente para generar prototipos de visualización de datos

a través de distintas colecciones de módulos. Éstos son extremadamente flexibles, dinámicos y amigables con el usuario. Sin embargo, no ofrecen una base adecuada para el análisis y la minería de datos, impidiendo al usuario el abordaje de tareas complejas. Además, tampoco facilitan la comunicación con otras aplicaciones y recursos externos. Ambientes como los descriptos son sistemas de software muy complejos, posiblemente distribuidos y con exigencias de tiempo real, y deben considerarse sujetos a frecuentes actualizaciones. Se puede afirmar entonces que la creación y el mantenimiento de estos sistemas plantean desafíos de performance, integrabilidad y modificabilidad a los desarrolladores de software.

A pesar de que estos sistemas no cumplen satisfactoriamente con las propiedades descriptas, sus características similares permiten afirmar que el área de la ***VI*** está madurando lo suficiente como para que sea posible desarrollar una ***arquitectura de referencia*** que sí facilite su cumplimiento. En la caracterización de la arquitectura de referencia se podrían aplicar avances producidos últimamente en el área de las Arquitecturas de Software dentro de la Ingeniería de Software, con el objetivo de asegurar que los sistemas de ***VI*** producidos a partir de la misma sean sistemas de calidad.

Una arquitectura de software provee un ***blueprin***t para un sistema, y el sistema debe construirse a partir de este ***blueprint*** [5,24,12]. El proceso de moverse desde el modelo del dominio, el análisis de usabilidad y la arquitectura a un sistema ejecutable trata con la creación de la estructura del sistema y su relación con la arquitectura, el uso de la arquitectura como base para crear una versión limitada del sistema, la incorporación de los componentes reusados y la conformación del sistema final para la arquitectura. Una arquitectura de referencia define un conjunto de elementos arquitectónicos que son compartidos y reusados entre los miembros de una comunidad de dominio para construir sistemas en ese dominio. La arquitectura de referencia satisface la necesidad de explicaciones abstractas de las diferencias y similitudes entre aplicaciones en el dominio. También, permite el intercambio de componentes entre desarrolladores y facilita el aprendizaje a los programadores que son novicios en el dominio.

La existencia de una arquitectura de referencia [3] para sistemas de ***VI*** permitiría la especificación de los componentes de hardware y software necesarios para su implementación, de acuerdo a los diferentes dominios de aplicación que han surgido recientemente, tales como las consultas a bases de datos, minería de datos, monitoreo de sistemas, etc. Se enfatizaría en la producción de sistemas modulares que ofrezcan funcionalidad dinámica, adecuada para llevar a cabo tanto tareas de análisis como de visualización de datos. Además, se posibilitaría que los sistemas resultantes sean capaces de conectarse a fuentes de datos externas, consideren restricciones temporales y elementos de historia, permitan distribuir las tareas de los distintos subsistemas de un sistema de visualización, y sean compatibles con los paquetes estándares gráficos y de visualización existentes.

En base a las necesidades planteadas se propone aplicar técnicas y herramientas recientemente desarrolladas en el área de las Arquitecturas de Software a los sistemas de VI, con el objetivo de crear una arquitectura de referencia que sirva para que los investigadores dispongan de un conjunto de conceptos comunes sobre los cuales obtener un mejor entendimiento de los problemas conceptuales y técnicos del área, y los desarrolladores puedan implementar ambientes que compartan una plataforma común. A partir de los componentes de esta arquitectura se podrá realizar una evaluación, testeo y comparación de las técnicas de visualización ya existentes, posibilitando el desarrollo de nuevas técnicas y facilitando la implementación de aplicaciones que las incorporen. Además, favorecerá la implementación de componentes reusables para integrar en futuras aplicaciones.

El proyecto abarca distintas etapas, que parten del estudio de los modelos de referencia de las aplicaciones existentes, incluyendo el pipeline de visualización que actualmente es el más extendido, para lograr el desarrollo de ambientes para dominios específicos tales como la visualización de información geográfica y el monitoreo de sistemas complejos. A partir de los resultados obtenidos, se procederá con la definición de una arquitectura de referencia para ambientes distribuidos de VI, y se implementará al menos un prototipo de sistema en alguno de los múltiples campos de aplicación.

Bibliografía

1. Baecker, R. M. and Buxton, W. A. S., Readings in Human-Computer Interaction. San Mateo CA.: Morgan Kaufmann Publishers, 1995.
2. Baeza-Yates, R., Ribeiro-Neto, B., Modern Information Retrieval, Addison Wesley, 1999.
3. Bass, L.; Clements, P.; Kazman, R.; Software Architecture in Practice, Addison Wesley, 1998.
4. Booch G.; Jacobson, I.; Rumbaugh, J.; The Unified Modeling Language User Guid,. Addison-Wesley, 1999.
5. Bosch, J.; Design \& Use of Software Architectures: Adopting and Evolving a product-line approach. Addison Wesley, 2000.
6. Card, S., Mackinlay, J., Shneiderman, B., Readings in Information Visualization - Using Vision to Think, Morgan Kaufmann, 1999.
7. Casavant, T.L., Singhal,M. Ed. Reading in Distributed Computing Systems. IEEE Computer Society Press, 1994.
8. ,G.F.; Dollimore, J. y T. Kindberg; Distributed Systems: Concepts and Design. 3rd Edition. Addison-Wesley, 2001.
9. Conrad, S., Ramos, J., Saake, G., Sernadas, C., Evolving Logical Specifications in Information Systems, pp. 199-228. In Chomicki, J., Saake, G., Logics for Databases and Information Systems, pp. 31-70. Kluwer Academic Publishers, 1998.
10. Chi, Ed H. A Framework for Information Visualization Spreadsheets. Ph.D. Thesis. University of Minnesota, Computer Science Department. March, 1999.
11. Derthick, M., Roth, S., Data Exploration across Temporal Contexts, Proceedings of Intelligent User Interfaces (IUI´00), pp. 60-67, New Orleans, January 2000.
12. Dikel, D.; Kane, D., Wilson, J., Software Architecture. Organizational Principles and Patterns, Prentice Hall Inc., 2001.
13. Dix, A., Finlay, J., Abowd, G., Beale, R., Human-Computer Interaction, Prentice Hall Europe, Second Edition, 1998.
14. Eick, S., Visual Discovery and Analysis, IEEE Transactions on Visualization and Computer Graphics, Vol. 6, No. 1, January'March 2000.
15. Godfrey, P., Grant, J., Gryz, J., Minker, J., Integrity Constraints: Semantics and Applications, pp. 265-306. In Chomicki, J.; Saake, G.; Logics for Databases and Information Systems, pp. 31-70. Kluwer Academic Publishers, 1998.
16. Loosley,C.; Douglas,F.; High Performance Client/Server, John Wiley,1998.
17. Mayhew, D., The Usability Engineering Lifecycle, Morgan Kaufmann, 1999.
18. Mc Clure, C., Software Reuse Techniques. Adding Reuse to the Systems Development Process. Prentice Hall Imn., 1999.
19. Meyer, J., Dynamic Logic for Reasoning about Actions and Change, pp. 281-314. In Minker, J.; Logic Based Artificial Intelligence. Kluwer Academic Publishers, 2000.
20. Nielsen, J., Usability Engineering, Morgan Kaufmann, 1993.
21. Nielson, G., Hagen, H., Müller, H. Scientific Visualization: Overviews, Methodologies and Techniques, IEEE Computer Society, 1997.
22. Raskin, J., The Human Interface, Addison Wesley, ACM Press, 2000.

23. Rumbaugh, J.; Jacobson, I.; Booch, G.; The Unified Modeling Language Reference Manual. Addison Wesley, 1999.
24. Shaw, M.; Garlan, D.; Software Architectures: Perspectives on an Emerging Disciplines. Prentice Hall, 1996.
25. Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N., Van de Velde, Wielinga, B., KnowledgeEngineering and Management. The CommonKADS Methodology. MIT Press, 2000.
26. Schroeder, W., Martin, K., Lorensen, B., The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics, Prentice Hall PTR, 1996
27. Shneiderman, B., Designing the User Interface, Addison-Wesley Publishing Company, 1998.
28. Spence, R., Information Visualization, Addison-Wesley, ACM Press, 2001.
29. Stasko, J., Domingue, J., Brown, M., Price, B. editores. Software Visualization: Programming as a Multimedia Experience. MIT Press, 1998.
30. Tanenbaum, A.S.; Modern Operating Systems. Prentice Hall, 2000.
31. Tufte, E., The Visual Display of Quantitative Information, Graphics Press, 1983.
32. Tufte, E.R., Envisioning Information, Cheshire, CT Graphics Press, 1990.
33. Tufte, E.R., Visual Explanations: Images and Quantities, Evidence and Narrative, Cheshire, CT Graphics Press, 1997.
34. Warmer, J., Kleppe, A., The Object Constraint Language, Addison Wesley, 1999.