Bidirectional Coupling - From 3D Field Simulation to Immersive Visualization Systems

Daniel Buendgens[2], Andreas Hamacher[2], Martin Hafner[1], Torsten Kuhlen[2] and Kay Hameyer[1]
1 - Institute of Electrical Machines – RWTH Aachen University
2 - Virtual Reality Group – RWTH Aachen University
Schinkelstrasse 4, D-52062 Aachen, Germany
E-mail: buendgens@vr.rwth-aachen.de, Martin.Hafner@IEM.RWTH-Aachen.de

Abstract—**The interactive exploration of complex simulation data have spurred a renewed interest in visualization techniques, because of their ability to give an intuitively clue for the interpretation of electromagnetic phenomena. This paper presents a methodology for a bidirectional coupling of VTK-based visualization systems to interactive and immersive visualization systems which are specially adopted for the handing and processing of large and transient simulation data. In this work, the coupling is demonstrated by the flexible virtual reality (VR) software framework ViSTA which is used by many national and international research groups.**

I. Introduction

Together with the development of computers, simulation software has attained nowadays a rather high level of sophistication. In many real-life engineering situations, the problem is no longer to be able to simulate, but rather to be able to interpret correctly and efficiently the huge amount of numerical information generated by the simulation. Visualization was regarded so far as an auxiliary task in finite element (FE) modeling that was limited to the servile representation of simulation results. But together with the increase in the complexity of simulation data, the interest in appropriate visualization systems is renewing nowadays. The general requirement for such systems is to enable a fast or even sub-real-time graphical representation of transient simulation data, irrespective of its data volume. For such scenario, specially adapted software concepts are required to fulfill such criteria. This paper presents the idea and the technical background of a bidirectional coupling of the VTK-based electromagnetic visualization tool Trinity.IVR and the VR-framework ViSTA.

II. Related Work

In many research projects the coupling of different software frameworks and software tools are an important issue. In [1] Schopfer et. al. present an integration platform (CHEOPS) for chemical process modeling and simulation. In this approach an overall integration framework is provided in which all simulation and visualization tools are integrated. Riedel et al. present a network-based coupling of the visualization tool VISIT and existing simulation algorithms running on a high-performance computing (HPC) system. Within the CONV project [2] the VISIT internal client - server architecture is used for handling the coupling process. In addition to these approaches for coupling different software tools there are also solutions for coupling algorithms to enhancing the capabilities of an existing software framework. In [3] Raffinn et. al. describe the coupling of algorithms for mass-spring systems and deformable objects with an existing VR Software framework to process distributed physical based simulations. Our approach builds upon the ideas of the presented papers and on our prior work presented in the section III.

III. Previous Work

Since 2001 the Institute of Electrical Machines (IEM) developed the command line based software tool iMoose.trinity [4] which provided a liberty of visualization algorithms for the electromagnetic field problems. The commands were realized in Python [5], so that users could assemble static visualization scenes in an individual Python script. Within the scope of [6] its enhancement iMoose.trinity.VR [7], [8] has been developed, which applied the Open-Source Visualization Tool KIT (VTK) [9]. In 2009 the IEM started the development of iMoose.trinity.IVR [10] and new electromagnetic visualization methods [11], [12]. This software tool combined the advantages of both former packages and is integrated in the daily use for visualization of FE data. Its concept is designed to add FE post-processing and processing facilities as long term research goals.

Beside the development of VR capabilities focusing on electromagnetics, the VR toolkit ViSTA [13] and the extension for flow phenomena ViSTA Flowlib [14] have been developed in the course of many research projects. ViSTA is a flexible and extensible software framework which is already productively used by many national and international research groups [15]. ViSTA has a scalable interface that allows its deployment in desktop workstations, small and large VR systems, like the 3D workplaces or CAVE-like systems. Hence, new modules and concepts are easily integrated and can be used on a wide range of different systems. Another important feature of ViSTA is its focus on advanced interaction techniques for full and semi-immersive environments like Cave or 3D workplaces.

IV. Goals and Methodology

The main goal of this work is to provide a framework for a flexible coupling between ViSTA and iMOOSE-Trinity, which is not regulated by the restrictions of certain middleware software, e.g. like CORBA [16]. The coupling methodology presented here, is only focusing on the VTK layer for realizing the coupling strategy, which generally allows an interconnection to any VTK-based visualization environment. Regarding the fact that ViSTA and iMOOSE-Trinity are both using VTK pipelines, new functionality can easily be integrated in both software tools during the coupling process.

A direct integration of ViSTA into iMOOSE-Trinity is not suitable because both pieces of software have their own data structure and different purposes. Both systems are expert systems with their own internal logic. The coupling must be regarded, not as a mere integration, but rather as an enhancement of functionality. The purpose of this coupling is to offer the possibility to link a simulation package with the whole range of immersive visualization systems, from 3D office systems up to cave-like systems.

The outstanding idea, which is in the focus of this work, is to implement such a linkage between software packages by a generalized network-based bidirectional coupling. Since techniques like making interactive cuts or seeding particles for flow (or flux line) representation are common in both systems such a mechanism mirrors all performed actions from one system to the other, so that one ends up with a consistent representation on both environments. This mirroring leads to an observation of all responsible state changes on each system, which are directly transfered to its counterpart.

The benefit in such an additional effort is to provide an extended electromagnetic processing and post-processing framework, where users employ the advantages of immersive VR-Systems, e.g. the interactive point or surface selection in 3D, by switching from the standard GUI-based finite-element environment directly to such professional VR environments.

For instance, in the case of leakage flux visualization, the interaction for navigating to the points of interest takes place in the ViSTA driven application feeding the iMOOSE.Trinity algorithm, which computes the corresponding closed magnetic flux lines. Since these algorithms provide VTK visualization data which are directly associated with graphical primitives, the coupling mechanism supplies these data objects on other display system. These type of interactions entails back-and-forth communications between ViSTA and iMOOSE.Trinity to achieve a seamless coupling integration, which can be utilized in classical, finite element based, design processes of electrical drives.

The allover design goal of mirroring functionality through different applications is far from being trivial and represents the major task of this project. This is even more true, when introducing vital side conditions, demanding minimal code changes in the FE simulation package to guarantee that further development features are compatible to the proposed coupling method. The technical background to do so is presented in section V, starting from the high level software design concept towards its practical implementation.

V. Software Architecture

A. High Level Concept

The software architecture of the coupling is deduced from the requirements. The identified requirements are:

- a coupling mechanism which directly relies on the VTK-framework to enable a linkage to any VTK-based application
- minimal code changes in the VTK-based application in order to achieve that no implementation overhead is generated for the further development of the application
- accumulation of necessary code changes on a single point of contact

Fig. 1 shows the high level concept of the ViSTA based coupling, showing a couplingmanager (CM) as central element of the communication relay. On each side the CM communicates state changes and arbitrary procedure calls with its counterpart listening on the base application. This manager is instantiated in each application and represents a single point of contact, using the facade pattern[17], for code changes and application enhancement as listed in the requirements above. To fulfill the requirement of minimal code changes within the application certain subparts of the program like the interaction subsystem and visualization content (see Fig. 1) are exposed via observer pattern to the CM, allowing the CM to observe their state changes without having any interdependence between them. In this context the observer do not need to have any application-specific knowledge from the observered application. Moreover, the ViSTA-CM detects all changes incoming from the VTK application also utilizing such a concept. Thereby, the CM encapsulates the complete coupling subsystems like network and state processing from the application itself, which are necessary to communicate internal changes with its counterpart (see Fig. 1) .

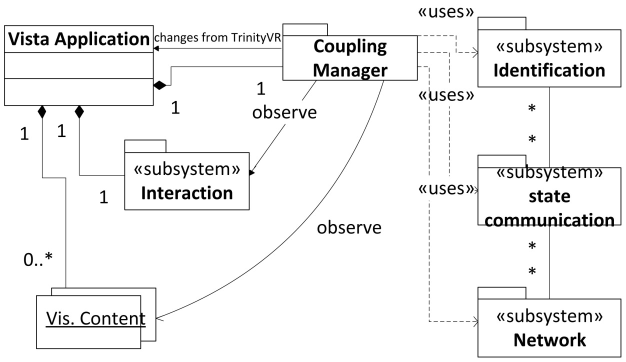

Fig. 1. Collaboration between the ViSTA Application and the Coupling Extension

To guarantee a further single usage of ViSTA and iMOOSE-Trinity the code changes are restricted to a minimum. In the coupled mode each application has to run a couplingmanager that encapsulates the layers in between and handles state changes generically. The applied facade pattern mentioned above is like a collection of all required interfaces in one class. In practice the couplingmanager provides a network interface to address the machine the coupled application is running on. Furthermore the application registers itself and the parts that should be synchronized between the applications. Modifications to most parts of the applications were made in derived classes. To encapsualte the different class instanciations of the coupled mode and the normal one, all instanciations are handled by a class-factory [17]. In VTK these factorys are already available and were only extended.

B. Concept of Remote Objects

This section will introduce the concept of logical objects called RemoteObjects like state data, the visualisation content or remote-procedure-calls(RPCs) [18]. Since the application coupling is targeting the visualisation and the user interaction, the low level coupling concentrates on exchanging the states of the visualisation content and the user interaction. For example both applications handle and visualize VTK visualization pipelines. One visualization pipeline describes one piece of visualisation content. But these visualisation

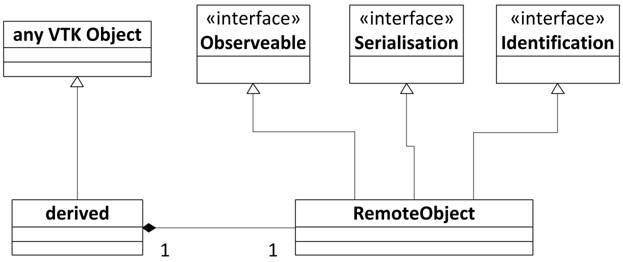

Fig. 2. RemoteObject Class Diagramm

contents are used in different windowsystems due to the fact that ViSTA is an application for heterogenous displaysystems. The fact that parts of the applications do the same (visualizing content) but partly work differently motivated the concept of RemoteObjects. These RemoteObject shall describe a logically equivalent part of each application with different implementations. RemoteObjects will allow a tight coupling and a generic state update mechanism over network on an per-object level. In this case a tight coupling stands for a communications interface between both applications to synchronize their states.

The concept for a generic handling of state updates requires minimal interface changes in the VTK application as shown in Fig. 2

To minimize the API changes for a VTK application all coupling mechanisms are implemented in derived classes from standard VTK objects. Each derived class aggregates a more generic RemoteObject which encapsulates the state handling including the state update processes. To enable the communication of state changes in an approprate granulation the Observer Pattern [17] is applied.

To minimize the network traffic of state updates, the existing VTK observer mechanism, which is generally available on each VTK object, is extended to provide a finer granularity. Finer granularity means in this case a higher level of detail to allow the observer mechanism to react only on the specific state changes on each VTK object. This results in a minimal communication overhead for large visualization objects, which are characteristic for FE computations (e.g. vector solution data), since only incremental changes are considered. This concept completely differs from the standard VTK state changes communication which is part of its pipeline principle resending modified objects completely. The VTK pipe concept is explained in [9]. Secondly, each RemoteObject implemented a serialisation and deserialisation interface to enable network communication (see Fig. 2). Thirdly, a mechanism for unique object identification was applied. These object IDs needed to be synchronized in both applications. Derivation is used instead of any other software design pattern, to enhance the class functionalities, also derivation preserves their normal usage and interfaces. In order to mirror the white rending window of VTK, one just has to derive this class together with the RemoteObject pattern described above. This modified VTK base class replaces the prior used rending window, so that each executed setter function calls the trigger for its normal VTK behavior and additionally communicates these changes to the corresponding RemoteObjects. The latter applies a standardized state update in the CM, so that all information is transferred to the ViSTA application.

C. Datafow Description

To explain the flow of information between both applications using each a couplingmanager synchronizing arbitrary RemoteObjects over network, the analogy of a layer based software architecture can be used. The software layout of the coupling mechanism consists of a six layers each describing a specific task in the state handling process, see Fig. 3. The first and last layer represent the application layers of both participating programs. These programs were extended to communicate a generic object state update.

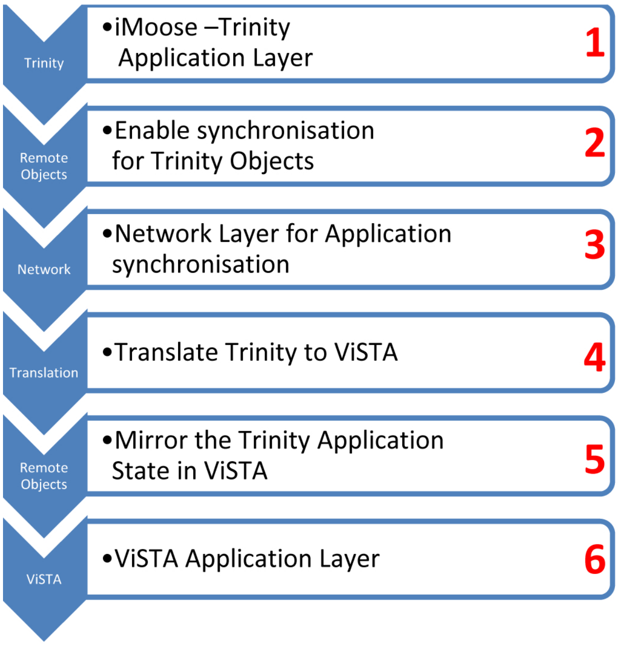
Trinity
•iMoose –Trinity Application Layer
1
Remote Objects
•Enable synchronisation for Trinity Objects
2
Network
•Network Layer for Application synchronisation
3
Translation
•Translate Trinity to ViSTA
4
Remote Objects
•Mirror the Trinity Application State in ViSTA
5
ViSTA
•ViSTA Application Layer
6

Fig. 3. State handling between iMOOSE-Trinity and ViSTA.

Each layer in between is part of the state transcription. The arrows associated with each layer represent the type of communication, whereas its communication direction between layer one and six can be reversed to ensure a bidirectional coupling interface. The synchronization of the iMOOSE-Trinity application and its runtime states on an per-object level is realized by RemoteObjects. RemoteObjects are automatically synced via Network.

Layer three represents the network communication. It is optimized for bidirectional synchronization states and considers the problem of parallel state manipulation. In most cases communication is handled asynchronously using Asynchronous Completion Token [19]. Semaphore locking and synchronous communication is used to preserve data integrity in vital parts like the unique object identification. The network connection is capable of direct socket connection on a single machine and communication over TCP/IP.

Since both applications use VTK techniques underneath, the overhead to enable them to communicate is limited to the interaction and RPCs in the majority of cases. The translation of logically equivalent tasks between the ViSTA and iMOOSE-Trinity, e. g. a camera update by an input device is handled in the fourth layer. But it is far from being trivial to realize this translation, because application-specific knowledge from each side are needed to succeed in this task. The implementations follow the Adapter pattern. The exchange of the visualization content is achieved by the VTK-Legacy format hence no conversation between graphical data at runtime is required.

In layer five the translation not of objects but of the interaction of classes is managed e.g. ViSTA supports multiple display systems, camera updates are more complex than in VTK and handled in more than one class. In this layer a efficent selection of logical equivalent objects is essential for a clearly laid out software architecture. Furthermore the application of the RPCs is done.

VI. Practical Issues

Even if professional immersive visualization systems are available today, such systems are seldom used as part of the design and development process of electrical drives to capture the influence of single, mostly local, phenomena with respect to design variations. The main reason for avoiding such techniques is rather simple and driven by practical issues.

The finite element simulation data for an appropriate visualization needs to be exported as graphical primitives or interpolation fields, yielding a static scene. The latter can afterwards be explored by different graphical techniques relying on these basic primitives. Since these kinds of data structures are comparable small to all additional information data necessary for FE computations, one yields a visualization system which is able to show a fast or even sub-real-time representation of transient simulation data. As drawback one lost any possibility to apply specific processing or post-processing routines, requiring the specific field solution together with its founding element shape functions, which are exclusively hosted by the FE software package.

Using the presented methodology for coupling both specific applications (FE-simulations package, ViSTA), all enhanced features available in immersive environments can be utilized as part of design and development processes for electrical machines. For instance, the scientist can use cave-like or Desktop-based VR systems. With the use of current tracking systems a user centered projection is feasible which allows a more intuitive view on the dataset from different perspectives. Since, advanced tracking systems provide the usage of advanced interaction techniques with 6 degrees of freedom, an interactive exploration of complex geometries and e. g. define cut plane for evaluation are more intuitively. Due to the data consistency, described in V, all further operations requiring finite element methods are still available, without facing the user with any kind of technical barrier, e.g. explicit data exchange, data conversion, etc.

This approach can help to integrate all beneficial enhanced VR features to FE packages as long term research goals.

VII. Conclusion

Efficient methods for the visualization of finite element solutions are essential for the evaluation of electromagnetic devices under research and development. This paper presents a generalized software architecture for the coupling of VTK-based graphical representations to use full and semi immersive visualization systems. The object state handling described in this paper allows a seamless interconnection between both systems without affecting and touching the other one. Since the presented work is currently in a prototyp-phase it is strongly necessary to carry out significant user studies with scientists dealing with electromagnetic phenomena in the next steps. The results the these user studies and the feedback from the scientists will be used in order to improve and enhance the provided coupling functionality.

References

[1] G. Schopfer, A. Yang, L. von Wedel, and W. Marquardt, "CHEOPS: a tool-integration platform for chemical process modelling and simulation," *International Journal on Software Tools for Technology Transfer*, vol. 6, no. 3, pp. 186–202, 2004.

[2] M. Riedel, W. Frings, S. Dominiczak, D. Mallmann, T. Eickermann, T. Gibbon, and R. Spurzem, "Interoperability of a collaborative online visualization and steering (COVS) framework using VISIT," Manchester, May 2007.

[3] J. Allard and B. Raffin, "Distributed physical based simulations for large VR applications," in *IEEE Virtual Reality Conference (VR 2006)*, Alexandria, VA, USA, pp. 89–96.

[4] C. Monzel and G. Henneberger, "Object-Oriented design of a visualisation tool on top of a FEM package," in *13th Conference on the Computation of Electromagnetic Fields*, vol. 2, Evian, France, 2001, pp. 216—217.

[5] G. van Rossum, "Python programming language," http://www.python.org, 2009.

[6] M. Schöning, *Virtueller Produktentwicklungsprozess für Elektromotoren*, 1st ed. Shaker, Aug. 2008.

[7] M. Schöning, M. Asbach, D. van Riesen, and K. Hameyer, "Visualising finite element solutions in virtual reality environments," in *CEM 2006*, Aachen, Germany, Apr. 2006.

[8] M. Schoning and K. Hameyer, "Applying virtual reality techniques to finite element solutions," *Magnetics, IEEE Transactions on*, vol. 44, no. 6, pp. 1422–1425, 2008.

[9] S. Aylward, S. Barré, A. Cedilnik, B. Hoffman, L. Ibáñez, B. King, K. Martin, K. Moreland, W. Schroeder, and A. Squillacote, "VTK 5.0 pipeline architecture," *The Kitware Source - Software Developer's Quarterly*, vol. Issue 1, pp. 4–6, Jul. 2006. [Online]. Available: http://www.kitware.com/products/thesource.html

[10] M. Hafner, M. Schoning, M. Antczak, A. Demenko, and K. Hameyer, "Methods for computation and visualization of magnetic flux lines in 3-D," *Magnetics, IEEE Transactions on*, vol. 46, no. 8, pp. 3349–3352, 2010.

[11] ——, "Interactive postprocessing in 3D electromagnetics," *Magnetics, IEEE Transactions on*, vol. 46, no. 8, pp. 3437–3440, 2010.

[12] M. Hafner, S. Böhmer, F. Henrotte, and K. Hameyer, "Interactive visualization of transient 3D electromagnetic and n-dimensional parameter spaces in virtual reality," in *21th Symposium on Electromagnetic Phenomena in Nonlinear Circuits*, Essen, Germany.

[13] T. van Reimersdahl, T. Kuhlen, A. Gerndt, J. Henrichs, and C. Bischof, "ViSTA: a multimodal, platform-independent VR-Toolkit based on WTK, VTK, and MPI," in *Fourth International Immersive Projection Technology Workshop (IPT2000)*, Ames, Iowa, 2000.

[14] M. Schirski, A. Gerndt, T. van Reimersdahl, T. Kuhlen, P. Adomeit, O. Lang, S. Pischinger, and C. Bischof, "ViSTA FlowLib - framework for interactive visualization and exploration of unsteady flows in virtual environments," in *Proceedings of the workshop on Virtual environments 2003 - EGVE '03*, Zurich, Switzerland, 2003, pp. 77–85.

[15] T. Duessel, H. Zilken, W. Frings, T. Eickermann, A. Gerndt, M. Wolter, and T. Kuhlen, "Distributed collaborative data analysis with heterogeneous visualisation systems," in *Eurographics Symposium on Parallel Graphics and Visualization*. Lugano, Switzerland: Eurographics Association, 2007.

[16] A. Pope, *The CORBA Reference Guide: Understanding the Common Object Request Broker Architecture*, 1st ed. Addison Wesley Publishing Company, Dec. 1997.

[17] E. Gamma, R. Helm, R. Johnson, and J. M. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, 1st ed. Addison-Wesley Professional, 11 1994.

[18] P. B. Hansen, Ed., *The Origin of Concurrent Programming: From Semaphores to Remote Procedure Calls*, softcover reprint of hardcover 1st ed. 2002 ed. Springer, 7 2011.

[19] D. Schmidt, M. Stal, H. Rohnert, and F. Buschmann, *Pattern-Oriented Software Architecture Volume 2: Patterns for Concurrent and Networked Objects*, 1st ed. Wiley, 9 2000.