Postprint of a Study on Resilience and Flexibility of Technical Systems in News Production

Authors: LING Yunyi

Date: 2025-07-09T00:00:00+00:00

Abstract

Objective: To investigate the resilience and flexibility of technical systems in news production systems, aiming to construct more flexible systems and technical solutions that adapt to and accelerate media production efficiency and creative capacity.

Methods: By employing templating, modularization, microservices architecture, and agile development methodologies, rapid response of technical systems is achieved.

Results: The application of these methods and technologies resolves bottlenecks and challenges in media production, thereby promoting efficiency and innovation.

Conclusion: These methods and applications can better unleash technological productivity, enhance media dissemination capacity and influence, and provide a reference for technical response in media systems.

Full Text

Preamble

Research on the Resilience and Flexibility of Technical Systems in News Production

(Hainan Nanhai Net Media Co., Ltd., Haikou, Hainan 570100)

Abstract

[**Objective**] This study investigates the resilience and flexibility of technical systems in news production, aiming to construct more agile systems and technical solutions that enhance media production efficiency and creative capacity.

[**Method**] The research employs templating, modularization, microservices architecture, and agile development methodologies to achieve rapid system response. [**Result**] The application of these methods and technologies resolves bottlenecks and challenges in media production, thereby promoting energy efficiency and innovation in media workflows. [**Conclusion**] These approaches and applications can better unleash technical productivity, enhance media dissemination power and influence, and provide valuable references for technical responsiveness in media systems.

Keywords: Special Events; Templating and Modularization; Resilience and Scalability; Flexibility and Extensibility; Media Convergence
CLC Number: G223
Document Code: A
Article ID: 1671-0134 (2025) 02-32-05
DOI: 10.19483/j.cnki.11-4653/n.2025.02.004
Citation Format: Ling Y. Research on the Resilience and Flexibility of Technical Systems in News Production [J]. China Media Technology, 2025, 32(2): 32-36.

Introduction

Since the birth of the internet, information sharing has accelerated dramatically. Web-based media emerged as a powerful force, and with the launch of rudimentary BBS platforms, online information dissemination and sharing became possible. The development of rich text and other related technologies further propelled internet growth, creating space for rich internet applications. Commercial media outlets such as Sina and Tencent News rapidly proliferated in the 1990s, prompting domestic institutional media to begin large-scale deployment of internet media platforms in the form of electronic newspapers and online television stations. After years of development, these have evolved into highly sophisticated and complex media networks.

The release of the "Guiding Opinions on Promoting the Integrated Development of Traditional and Emerging Media" and the "Opinions on Accelerating In-Depth Media Integration Development"marked a new era for the media industry. In this developmental process, as we review history and look to the future, information technology and internet media have been inseparable. Technological advancement has propelled media development forward, while further media development has placed new demands on technology. The two are dialectically unified and mutually supportive. This paper focuses on how the resilience of technical systems during media development impacts news production, using the implementation of news special features as an entry point to explore how superior technical solutions can address insufficient technical elasticity and constraints on news production.

1. Manifestations of Technical Constraints

Hainan Nanhai Net Media Co., Ltd. is an online media subsidiary of Hainan Daily Newspaper Group. Developed gradually from the foundation of Hainan Daily' s electronic newspaper, it has become the "Net" in Hainan Province' s "One Newspaper, One Net, One Station" media structure and represents the official online media platform of Hainan Province, operating under the brand "Nanhai Net." Throughout its development, the company' s technical needs have taken various forms.

In the early stages, to address technical system requirements, the primary approach involved purchasing CMS systems, supplemented by in-house development and secondary development of open-source CMS platforms such as early versions of PHPCMS and DedeCMS [1]. Most purchased CMS systems were primarily designed for backend support. During the PC era, frontend support mainly focused on generating static HTML for SEO-driven traffic. However, due to insufficient in-house development capabilities, this technical architecture gradually became inadequate for supporting additional functionalities, particularly dynamic features centered on interactivity. This frequently resulted in a disconnect between news planning and technical requirements. When custom development was procured from vendors, the economic and time costs often exceeded the temporal demands of news itself. The rapid response required by the timeliness of news events created a conflict with development speed, ultimately leading to downgraded implementations based on existing system capabilities and suboptimal performance in news expressiveness and dissemination power.

To address this issue, templating solutions gradually emerged. These involved pre-configuring commonly used templates in the CMS backend, primarily for news special topics and page style templates. Template implementation relied mainly on tag systems [2], which associated tags with data. During page generation, regular expression matching and replacement were used to render data into HTML pages [3]. While templating aimed to solve development cost and rapid response issues, the relatively fixed styles—essentially static news display in building-block form, mostly limited to changing header images—resulted in rigid page layouts that constrained news expressiveness. Consequently, template utilization rates remained low, and editors and news production staff remained dissatisfied with technical response speeds. Moreover, templates primarily featured static information display with minimal dynamic functionality and poor interactivity, weakening the dissemination power and influence of special features and news content. For instance, covering events like space launches or sports games required interactivity to create a sense of participation—something templating could not adequately address and still necessitated custom development, thus returning to the original problem.

2. Evolution of CMS and Templating Solutions

2.1 Commercial CMS and Templating

As previously mentioned, after purchasing commercial CMS systems, Nanhai Net gradually accumulated numerous templates through procured custom development services, selecting different templates based on news themes. Template implementation primarily relied on tag systems, which still required varying degrees of technical involvement during page production. This consumed the technical team' s energy in creating numerous special features and activity pages. To resolve this contradiction, Nanhai Net established news production workflows that mandated the use of pre-configured templates for non-critical special features and activities, supplemented by minimal artistic design. This approach significantly alleviated the conflict between technical constraints and news production, improving news timeliness, dissemination power, and influence to some extent. However, underlying system constraints remained unresolved, limiting other functionality implementations and leaving insufficient elasticity, particularly for non-templated content.

Since the purchased CMS served as the core news production system and lacked adequate flexibility, Nanhai Net developed a small in-house CMS and employed secondary development of open-source platforms like PHPCMS, DedeCMS, and WordPress to support dynamic features and secondary channels. Functions such as comments, message boards, and specialized channels for automobiles, real estate, and health partially alleviated technical constraints and improved technical responsiveness. However, this created a complex situation of coexisting technical systems. With limited team resources, maintaining numerous siloed systems while producing thousands of special features and activities annually incurred substantial technical debt. As the number of special features and systems grew, integration costs became prohibitively high, ultimately sacrificing opportunities and capacity for iteration. Meanwhile, severe fragmentation and disconnect between technical systems persisted. While the core CMS lacked controllability, the in-house and secondary-developed systems offered some technical control and flexibility, but the "weakest link" principle meant constraints remained inadequately addressed. Additionally, issues with code standardization and security in the PHP-based technical system gradually became prominent during development. These technical constraints have accompanied Nanhai Net' s development to this day, as it continues striving for optimal solutions.

2.2 In-House CMS and Templating

To further address the flexibility limitations of commercial CMS, in-house development became an inevitable technical path for Nanhai Net. To rapidly respond to technical requirements, the company adopted the popular PHP language to build its own CMS system. Based on its technical capabilities and actual needs, the system focused on designing the most frequently used functions in news production, maintaining a lightweight architecture to prevent decreased iteration

speed due to excessive functionality.

The initial system design emphasized lightweight architecture. First, only essential functions were implemented: permission management, user management, channel management, news management, comment management, and statistical management—essentially a semi-finished system that facilitated easier extension. Second, code was uniformly encapsulated by functional modules with clear distinctions to achieve modular design. Public calling modules were kept as lightweight as possible with minimal functionality to enable easy extension. Finally, the approach of using code snippets in news production was introduced. Combined with templating concepts, storing small code snippets in the database allowed for rapid adaptation to extended development requirements with minimal modifications.

Database design did not strictly adhere to normalization principles. To facilitate extension and implementation, some table and field redundancies were retained, sacrificing some database performance and space but better balancing code extension speed. To better address rapid templating development, Nanhai Net customized a set of template tags (see Figure 1 [Figure 1: see original paper]) on its self-developed CMS and used Dreamweaver extensions to load these tags into the software via XML. Functionality could be quickly implemented by simply clicking the corresponding tag buttons—an approach inspired by industry solutions [4].

The self-developed 1.0 system solution compensated for constraints imposed by purchased CMS, alleviating the crisis for a time. However, cybersecurity issues became increasingly prominent. The loosely designed system had too many vulnerabilities and was easily compromised by hackers, with repair costs being extremely high. Meanwhile, the technical system became outdated, and technical constraints re-emerged.

2.3 Open-Source CMS

To supplement insufficient in-house development capacity, Nanhai Net historically used open-source CMS platforms like PHPCMS and DedeCMS for many sub-sites and client-facing websites. These systems also employed template solutions based on tags, using regular expression replacement to render static HTML. The issues faced during secondary development were similar to those with commercial CMS: finished systems had relatively lower flexibility, secondary development was difficult, performance degraded rapidly with large data volumes, optimization was challenging, and flexibility in news production was hard to extend upon the existing system model. While these systems completed some news support by leveraging their strengths, system resilience and extensibility remained insufficient, and cybersecurity issues were also significant.

2.4 Mobile Internet Era Solutions

The rapid development of mobile internet swept through the entire media industry. Changes in media ecology and user behavior, characterized by the "fast eats slow" nature of online media, created unprecedented pressure on traditional media like Nanhai Net. Technical constraints became increasingly apparent as business required rapid transformation, yet technical system elasticity was insufficient and development capacity could not meet the needs of change.

The emergence of new technologies in the mobile internet era, particularly APP and H5 technologies, drove the development of the technical ecosystem and reshaped the media industry. Problems that remained unsolved in the PC era were gradually resolved in the mobile internet era. The approach to templating changed with the introduction of rich text visualization supported by H5, dramatically reducing template usage difficulty. Ordinary editors could quickly complete special features and templates through building-block approaches without technical staff involvement. As the focus shifted to mobile-first, functional requirements for PC declined. Meanwhile, mobile device performance and screen size limitations led to quantified page designs that emphasized refinement over visual impact, alleviating pressure from the PC era' s focus on visual shock and enabling more refined user experiences.

To address news business elasticity, Nanhai Net implemented modular and flexible UI configuration solutions on its APP to rapidly respond to elasticity in news production. To better support technical needs, Nanhai Net assembled a development team for a lengthy in-house development process, adopting a microservices architecture [5] and implementing systems using both PHP and JAVA programming languages.

The PHP-based system primarily responded to rapidly changing requirements, such as support for websites, mini-programs, and lightweight APPs, leveraging PHP' s lightweight and fast characteristics. In actual news production, the technical team frequently received extremely urgent requirements, particularly for highly time-sensitive news where technical response time was required at the hourly level. The JAVA solution primarily supported core business components, especially systems requiring high performance, stability, and security, with long-term iteration needs and acceptable iteration cycles from a technical perspective. Development processes were designed to be agile, and system scale was controlled to avoid reduced elasticity from excessive size. As traffic was diverted to third-party platforms, self-operated platforms did not require particularly large and complex systems for support.

To meet the efficiency requirements of custom development and cybersecurity demands, Nanhai Net developed a lightweight system using PHP. This system was more suitable for the mobile internet era, retaining the lightweight characteristics of the PC-era self-developed system while implementing high-frequency functions such as user systems, permission systems, news publishing, and column management, preserving extensibility as much as possible. To better adapt

to the rich frontend trend, a frontend-backend separation design was adopted, decoupling backend API capabilities from frontend UI to further 解除 coupling constraints while supporting multi-terminal scenarios including APP, H5, and PC.

The dependency on SEO in the PC era had been a key design consideration. However, the reduced reliance on SEO in the mobile internet era provided conditions for new directions in system design and platform operation.

3. Technical Implementation

3.1 High-Elasticity Technical System Implementation Based on PHP

From a software engineering perspective, improving development efficiency focuses on project management processes and specific technical solutions. Processes should adopt agile development as much as possible, and technical systems should employ low-constraint approaches. To more quickly address technical elasticity issues, Nanhai Net' s technical team began developing an in-house CMS system, inheriting and building upon the experiences and lessons accumulated by previous technical staff. After thorough research, THINKPHP was adopted as the underlying framework, though not fully utilizing MVC design. Instead, a frontend-backend separation approach was implemented, with the backend providing only APIs and corresponding capabilities while the frontend requested data via AJAX. In practice, using THINKPHP as the foundation—while adhering to cybersecurity design principles—mitigated cybersecurity risks. The underlying support improved development efficiency, code standardization followed 底层设计 principles, avoided code and architecture chaos, and retained extensibility to support elasticity and code reuse.

The following example uses Nanhai Net's "Hainan Political Consultation"system to illustrate lightweight and elastic design.

[Figure 2: see original paper] Lightweight CMS Backend

3.1.1 Permission Management System For simplicity, the permission management system only implemented basic functions including three-tier review processes, role management, permission management, and administrator management. The design was kept as simple and loosely coupled as possible to facilitate subsequent extension.

3.1.2 Membership System The membership system only implemented the most basic user management functions, such as ID, mobile phone number, and login time—only the necessary fields. In accordance with national laws and regulations, user information collection was minimized to ensure security while providing sufficient expansion space for the system.

3.1.3 Content Management System Content management only implemented core content element functions including column management, content management, and material management. Data field design was also streamlined as much as possible to ensure extensibility.

3.1.4 Statistical System Statistics only covered necessary core data such as content volume, user count, and click-through rates.

3.1.5 Summary Since the mobile internet era, this lightweight CMS has supported and extended numerous business systems for Nanhai Net. Particularly in the past, the company undertook many highly customized businesses, yet system elasticity was insufficient to support them. The basic functions of this lightweight CMS are the most frequently used, providing relatively good guarantees for business extensibility, reusability, and security.

To address infrastructure elasticity issues, Nanhai Net adopted public cloud solutions. Unstructured data storage such as video, images, and audio used OSS object storage [10], while load balancing [11] and databases utilized PAAS-layer cloud capabilities as much as possible, simplifying operations and maintenance while providing reliability. For third-party APIs, Nanhai Net required vendors to cloudify their capabilities with as fine a granularity as possible, given that smaller granularity facilitates easier extension and reduces coupling constraints.

3.2 High-Elasticity Technical System Implementation Based on JAVA

JAVA technology systems, evolving from J2EE through SSH framework-based MVC architecture and SOA architecture to current microservices architecture, have greatly promoted software system iteration rates, especially when combined with frontend-backend separation architecture. Through reasonable module and service splitting and well-designed APIs, system elasticity can be maximally enhanced to facilitate iterative development and respond to business needs. Based on JAVA microservices architecture, Nanhai Net designed and developed its core support system, the "Haichao Smart Media Cloud" system.

3.2.1 Architecture System Currently, microservice frameworks represented by Spring boot have matured. After research, Nanhai Net's technical team chose Spring boot as the microservices foundation architecture system. Building upon agile development, the entire system was split into 31 subsystems (services) according to microservices design principles. Services interact via API calls using OpenJDK as the JAVA runtime environment. With cloud computing development, microservices and containers have become good partners, but based on actual needs, Nanhai Net did not directly adopt container solutions all at once. Instead, according to actual requirements, cloud hosts were used as the microservices runtime environment, kept as clean as possible [6].

At the database level, to reduce program complexity, MySQL database's master-slave mode was adopted, with write operations handled by the master instance

and read operations by slave instances. To address performance issues, K-V memory database Redis was used to cache high-frequency data in memory, reducing MySQL database pressure and improving data loading speed. As news data primarily consists of text, to solve full-text indexing and large data storage, Elasticsearch [7] was introduced, improving search performance and full-text indexing capabilities while alleviating MySQL database pressure from data volume growth. Frontend-backend separation architecture [8] was also adopted, with the frontend introducing the Vue framework [9] to call backend APIs and incorporating Vue-based page rendering mechanisms to improve frontend page performance and SEO.

[Figure 3: see original paper] Spring Boot Architecture Diagram

3.2.2 Layout Management System The layout management system was designed by Nanhai Net according to its technical elasticity requirements as a modular system adapted for APPs. Based on news reporting needs, it enables sequential adjustment, display/hide functionality, and style adjustments for APP page blocks, better adapting to scenario-based needs in news production. Multiple styles were designed specifically to avoid rigid APP interface layouts and aesthetic fatigue. Visualization drag-and-drop technology was employed for sorting, improving the user experience for system operators.

[Figure 4: see original paper] Layout Design Backend Screenshot

3.2.3 Site Management System To better adapt to subsite expansion, Nanhai Net designed a site cluster management system to facilitate subsite construction and extension. Site setup can be completed by simply adding a domain name through the backend configuration, with the API reading data according to different parameters.

3.2.4 Style Theme Management System To adapt to extension requirements for special time points in news production, Nanhai Net designed multiple themes in the system, applying templating concepts to scenarios such as festivals and events where different themes could be used as needed.

3.2.5 Other Elastic Design Features For highly flexible functions such as points, comments, and likes, Nanhai Net designed rule-based configurations in the backend as much as possible, enabling adaptation to current changing needs in news production through parameter configuration.

4. Key Technical Points

Through years of exploration in addressing news production system elasticity, Nanhai Net has gradually accumulated experience. The use and exploration of templating have to some extent bridged the gap between technical response

and business requirements. Combined with in-house development, Dreamweaver plugins were used to simplify templating work.

The application of visual drag-and-drop and visual design has improved work efficiency while enhancing the user experience for editors and other system users, reducing error rates from manual code or content modifications. The ThinkPHP framework provides foundational MVC capabilities with high-quality 底层代码 and commercial support, offering a good domestic ecosystem that enables high development efficiency and avoids excessive low-level implementation. To ensure system extensibility, Nanhai Net' s upper-layer systems were actually designed as semi-finished products, a design philosophy that better reduces the high cost of changes in finished systems and makes iteration relatively easier with loosely coupled design.

The Spring boot microservices architecture splits the system and uses API calls for data sharing and inter-service interaction, reducing subsystem or module size and improving iteration speed. With a mature technical ecosystem, it provides good security and performance guarantees. After splitting the system into over 30 services, good extensibility and reusability were retained. Meanwhile, the stability and performance of the JAVA environment were gradually demonstrated in the project, with services rarely becoming unavailable after system launch.

As the world changes, the technical ecosystem also evolves. Problems that previously could not be solved now find solutions with new technologies and tools. Simultaneously, changes in the business ecosystem create new opportunities for problem-solving. For instance, issues with templating application levels and extensibility that existed in the PC era were alleviated in the mobile internet era through application lightweighting and changed experience criteria. With small screens and reduced emphasis on visual impact, many previous burdens were lightened and gradually resolved, allowing focus on new problems in new scenarios. This paper, through a small vertical entry point, reproduces the technical challenges Nanhai Net faced as a representative of traditional media in solving technical problems, reflecting the path of technical development in traditional media and serving as a microcosm of technical transformation during media convergence.

References

[1] Xu X, Zhu C. Design and Implementation of News System [J]. China New Communications, 2022(18): 61-63.
[2] Wang X, Chen X, Wang H, Wang W. Automatic Extraction of Key Information from News Webpages Based on Tags and Block Features [J]. Journal of Shandong University (Natural Science), 2019(3): 67-74.
[3] Chen J. Research on Web Information Extraction Technology and Application [J]. Digital Technology and Application, 2017(9): 114, 118.
[4] Ma X, Li H. A Web Information Extraction Method Combining DOM Tree

Tag Paths and Line Block Density [J]. Intelligent Computer and Applications, 2017(4): 13-16, 20.
[5] Zhou L. Spring Cloud and Docker Microservices Architecture in Action [M]. 2nd ed. Beijing: Electronic Industry Press, 2018: 1-5.
[6] Li W, Zhang H. Research on Media Data Center Infrastructure Construction Based on Cloud Computing Technology [J]. China Media Technology, 2014(14): 62-66.
[7] He Y, Zhang Q. Design and Implementation of Private Cloud API Gateway and Monitoring System Based on Kong and Elasticsearch [J]. Computer Applications and Software, 2022(11): 136-140.
[8] Luo Q, Wu Y, Fei Y. Design of Journal Editorial System Based on AJAX Technology [J]. Electronic Technology and Software Engineering, 2021(2): 189-190.
[9] Piao L. Deep Dive into Node.js [M]. Beijing: People' s Posts and Telecommunications Press, 2014.
[10] Liu Y. Research and Implementation of Load Balancing Technology in Distributed Block Storage Systems [D]. Wuhan: Huazhong University of Science and Technology, 2016.
[11] Gao W, Ren B, Wei H, Zhou C. Exploration and Application of Distributed Object Storage in Editorial and Media Convergence Platforms [J]. China Media Technology, 2023(6): 104-107, 158.

Author Profile: Ling Y (1986-), male, from Tianshui, Gansu, Senior Engineer. Research interests include media convergence, technical management, project management, software engineering, and cybersecurity.

(Responsible Editor: Li Jing)

Note: Figure translations are in progress. See original paper for figures.